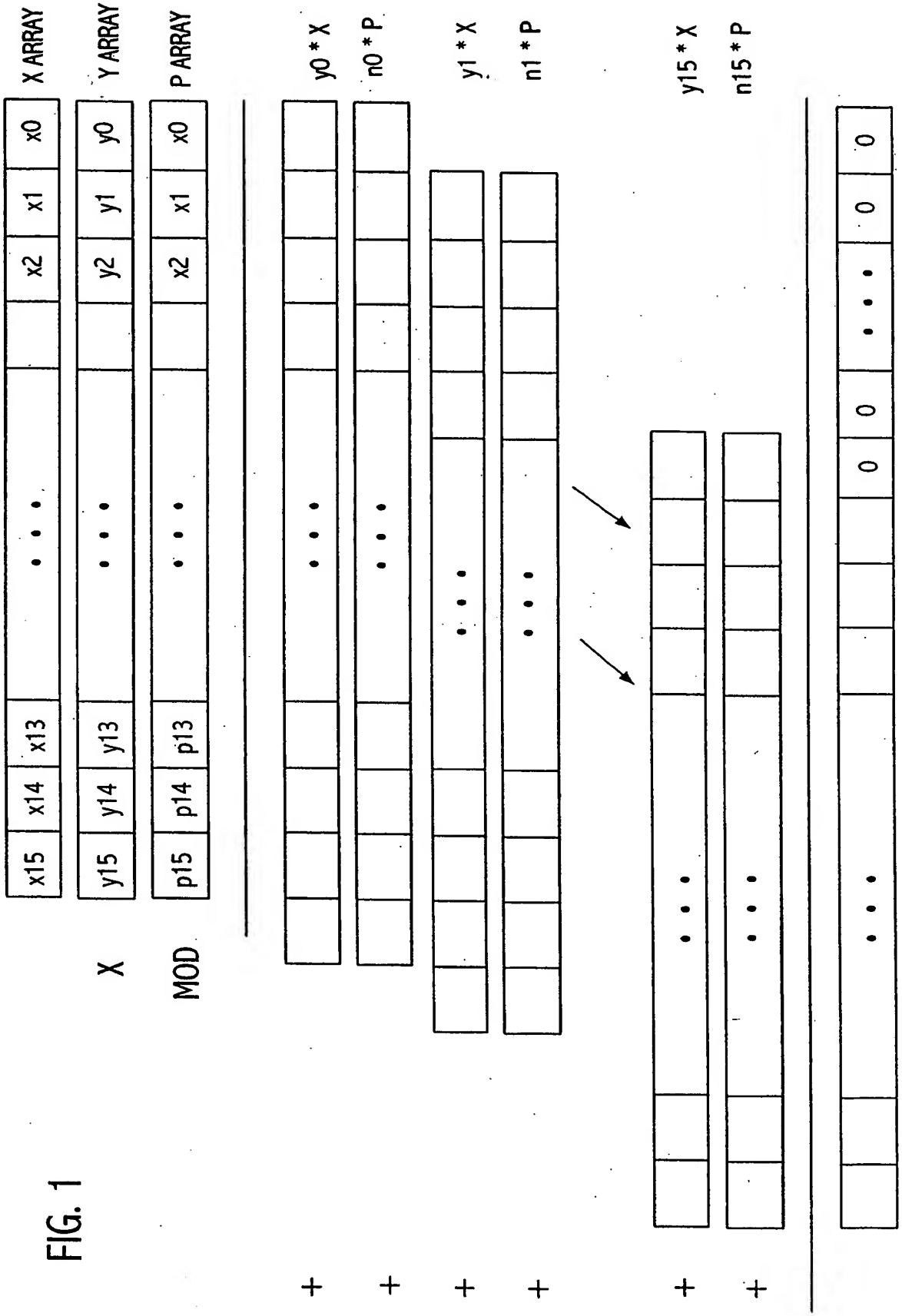

X ARRAY
Y ARRAY
P ARRAY
x0
x1
x2
x13
x14
x15
y0
y1
y2
y13
y14
y15
p13
p14
p15
X
MOD
y0 * X
n0 * P
y1 * X
n1 * P
y15 * X
n15 * P
0
+

FIG. 1

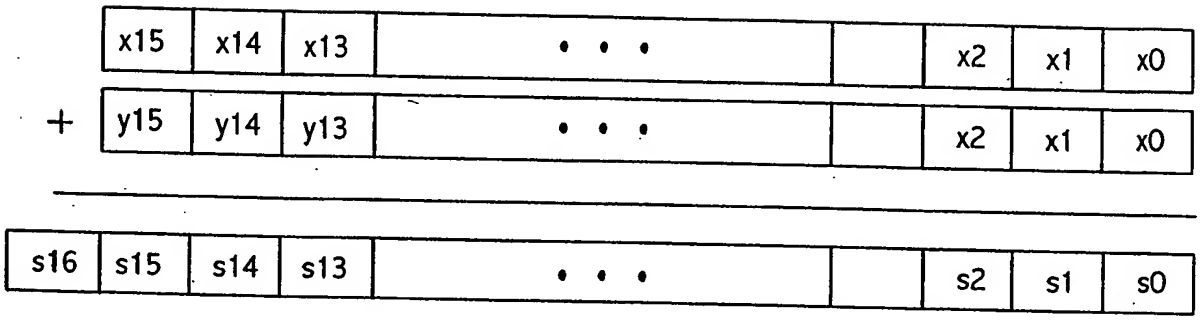


FIG. 2

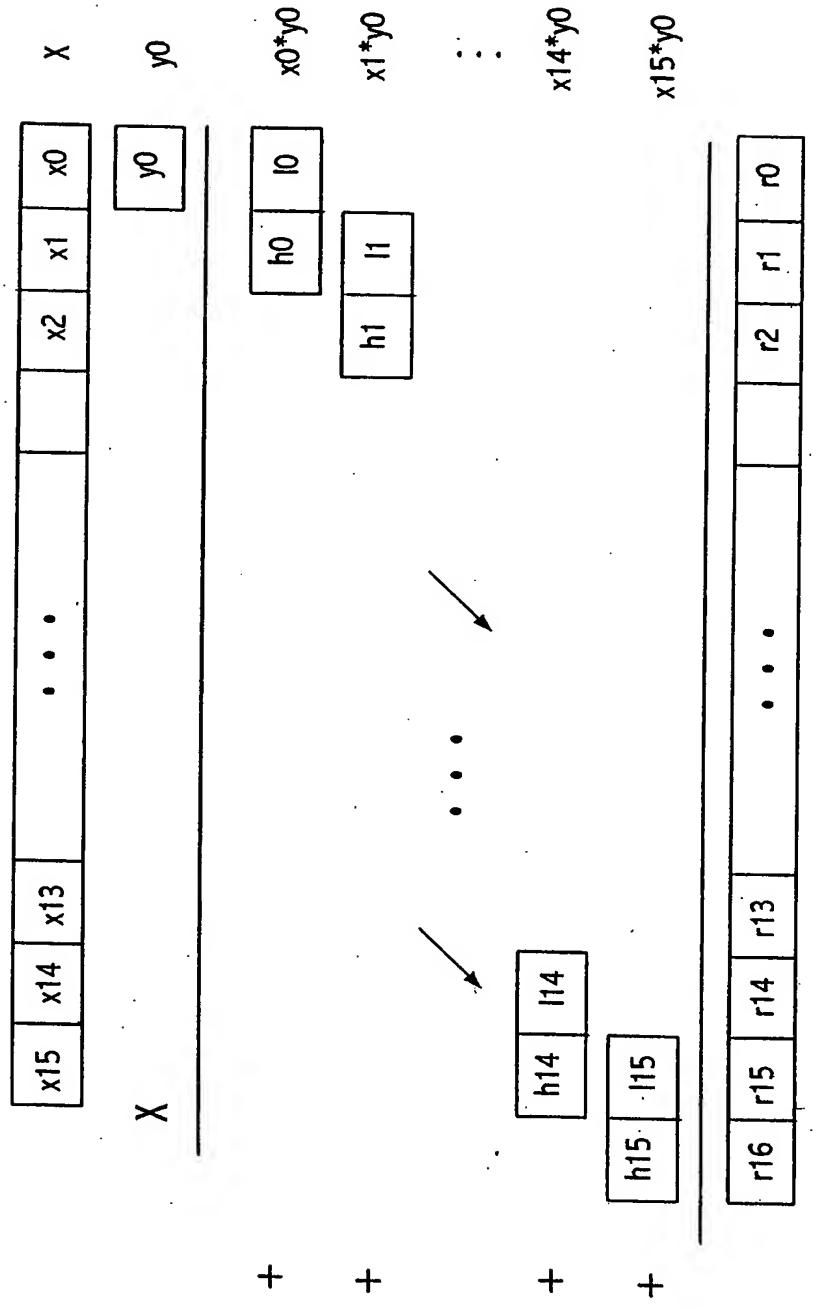

X
y0
x0*y0
x1*y0
x14*y0
x15*y0
x15
x14
x13
x2
x1
x0
y0
h0
l0
h1
l1
h14
l14
h15
l15
r16
r15
r14
r13
r2
r1
r0
+
+
+
+

FIG. 3

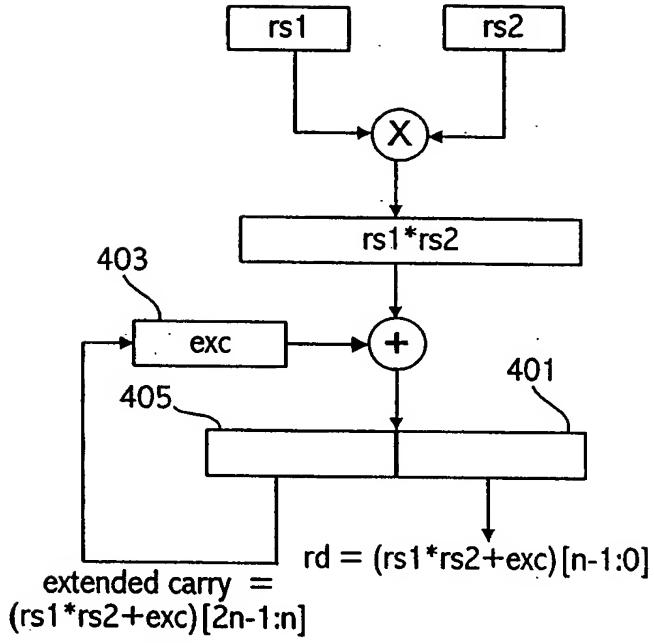

rs1
rs2
X
rs1*rs2
403
exc
+
401
405
rd = (rs1*rs2+exc)[n-1:0]
extended carry =
(rs1*rs2+exc)[2n-1:n]

FIG. 4A

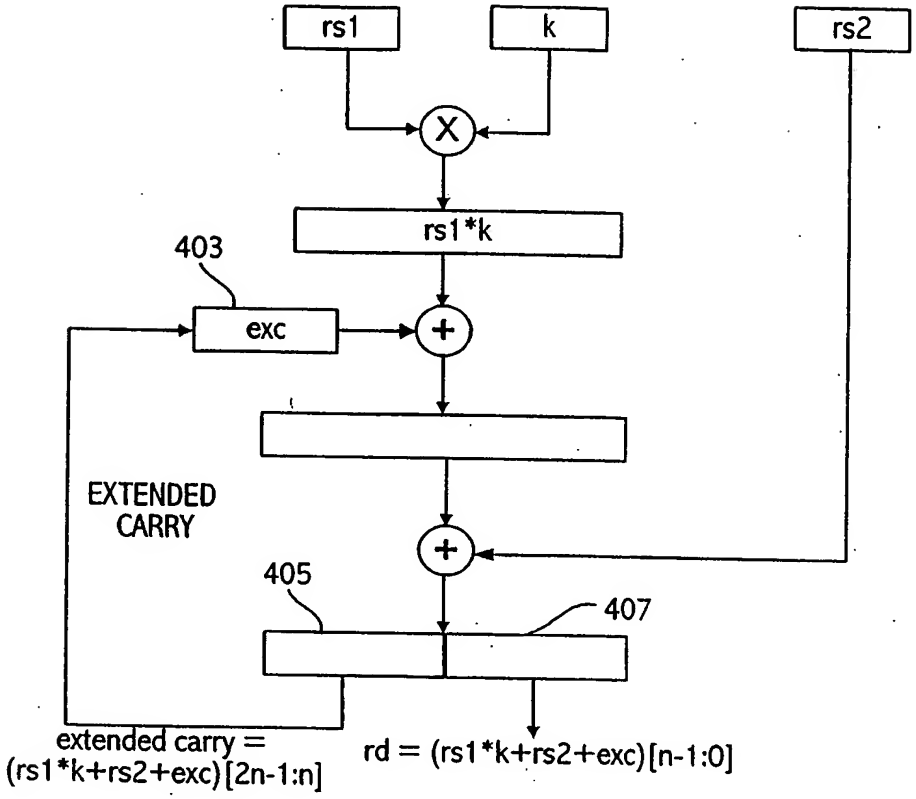

rs1
k
rs2
X
rs1*k
403
exc
+
EXTENDED
CARRY
+
405
407
extended carry =
(rs1*k+rs2+exc)[2n-1:n]
rd = (rs1*k+rs2+exc)[n-1:0]

FIG. 4B

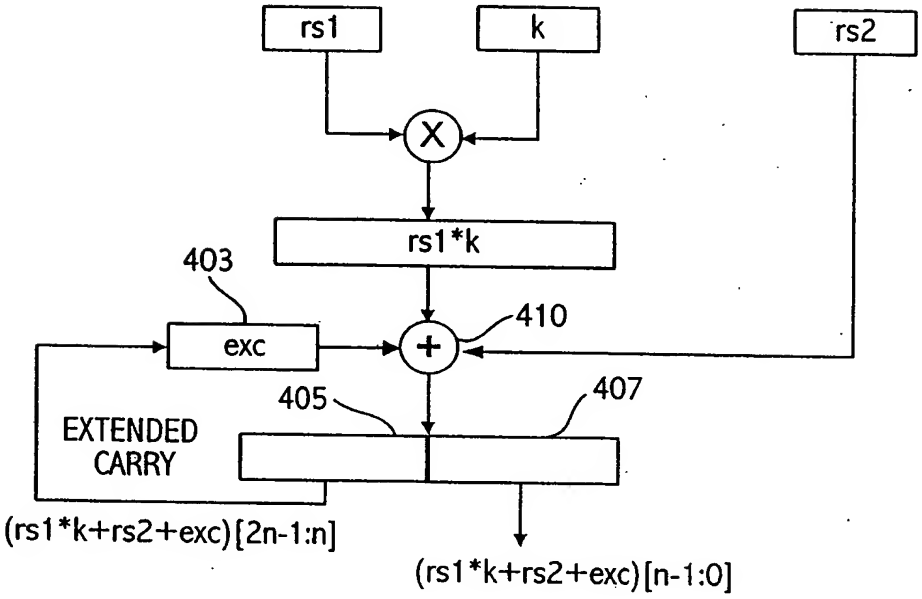

rs1
k
rs2
X
rs1*k
403
exc
410
+
405
407
EXTENDED
CARRY
(rs1*k+rs2+exc)[2n-1:n]
(rs1*k+rs2+exc)[n-1:0]

FIG. 4C

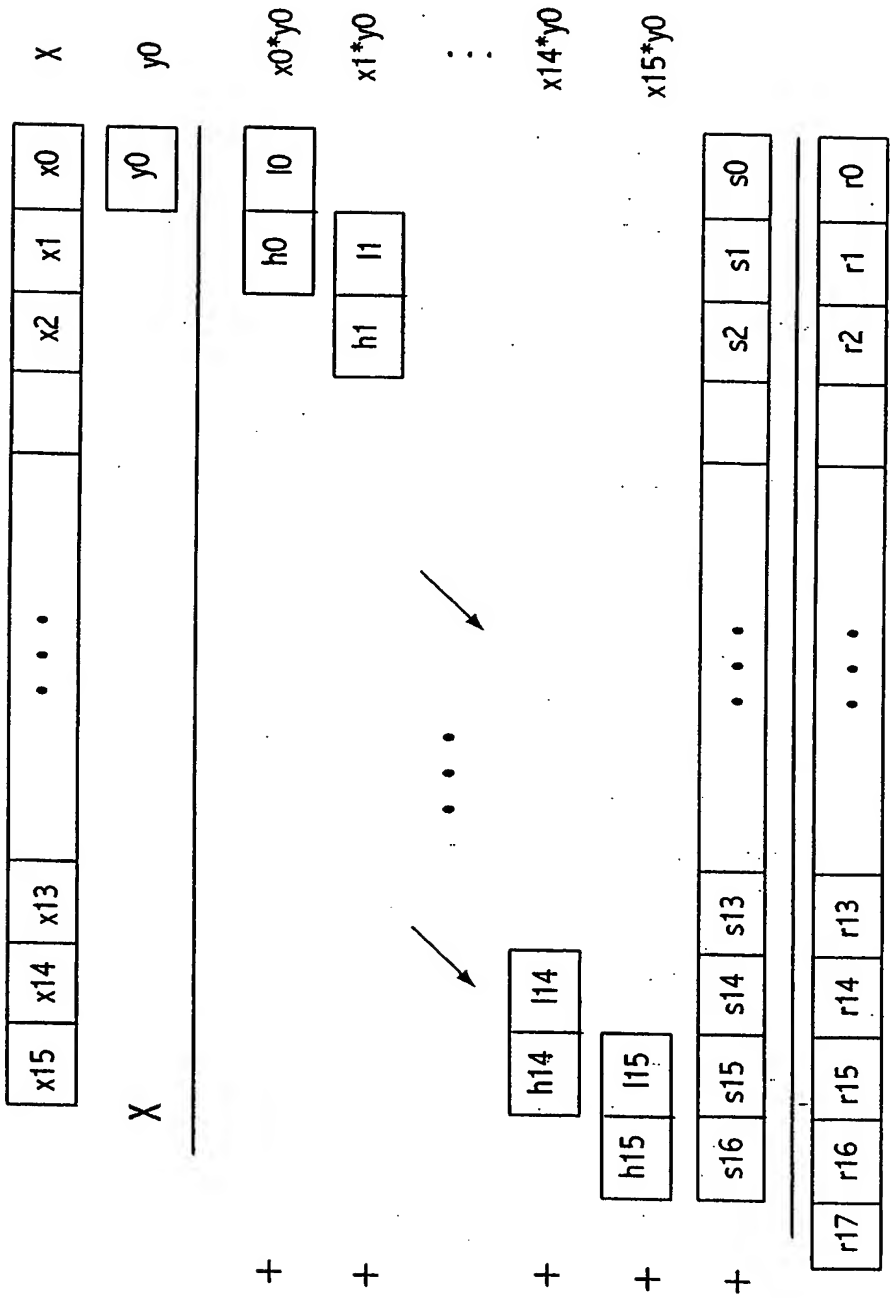

X
y0
x0*y0
x1*y0
x14*y0
x15*y0
x0
x1
x2
x13
x14
x15
y0
h0
l0
h1
l1
h14
l14
h15
l15
s0
s1
s2
s13
s14
s15
s16
r0
r1
r2
r13
r14
r15
r16
r17
+

FIG. 5

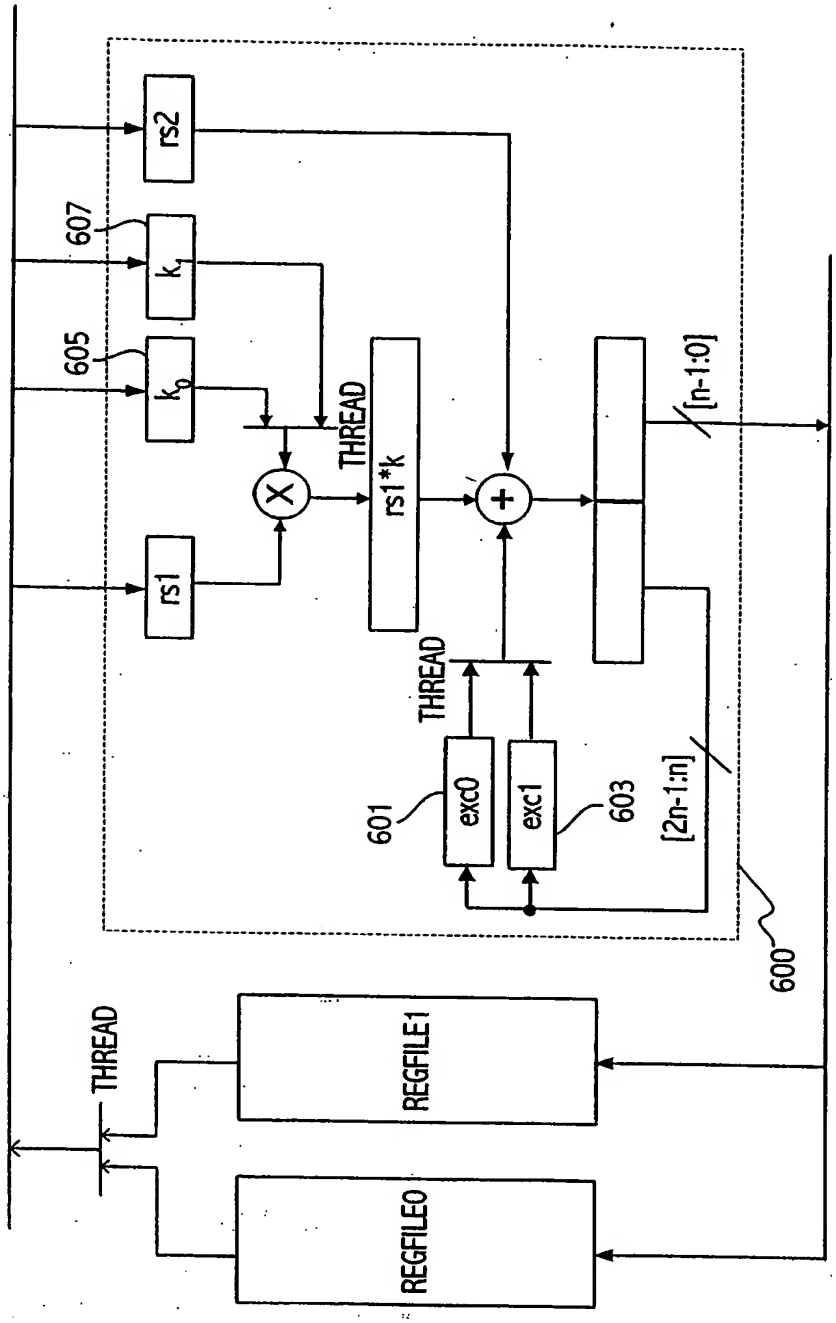

rs2
607
k_1
605
k_0
rs1
THREAD
rs1*k
[n-1:0]
THREAD
601
exc0
exc1
603
[2n-1:n]
600
THREAD
REGFILE1
REGFILE0

FIG. 6

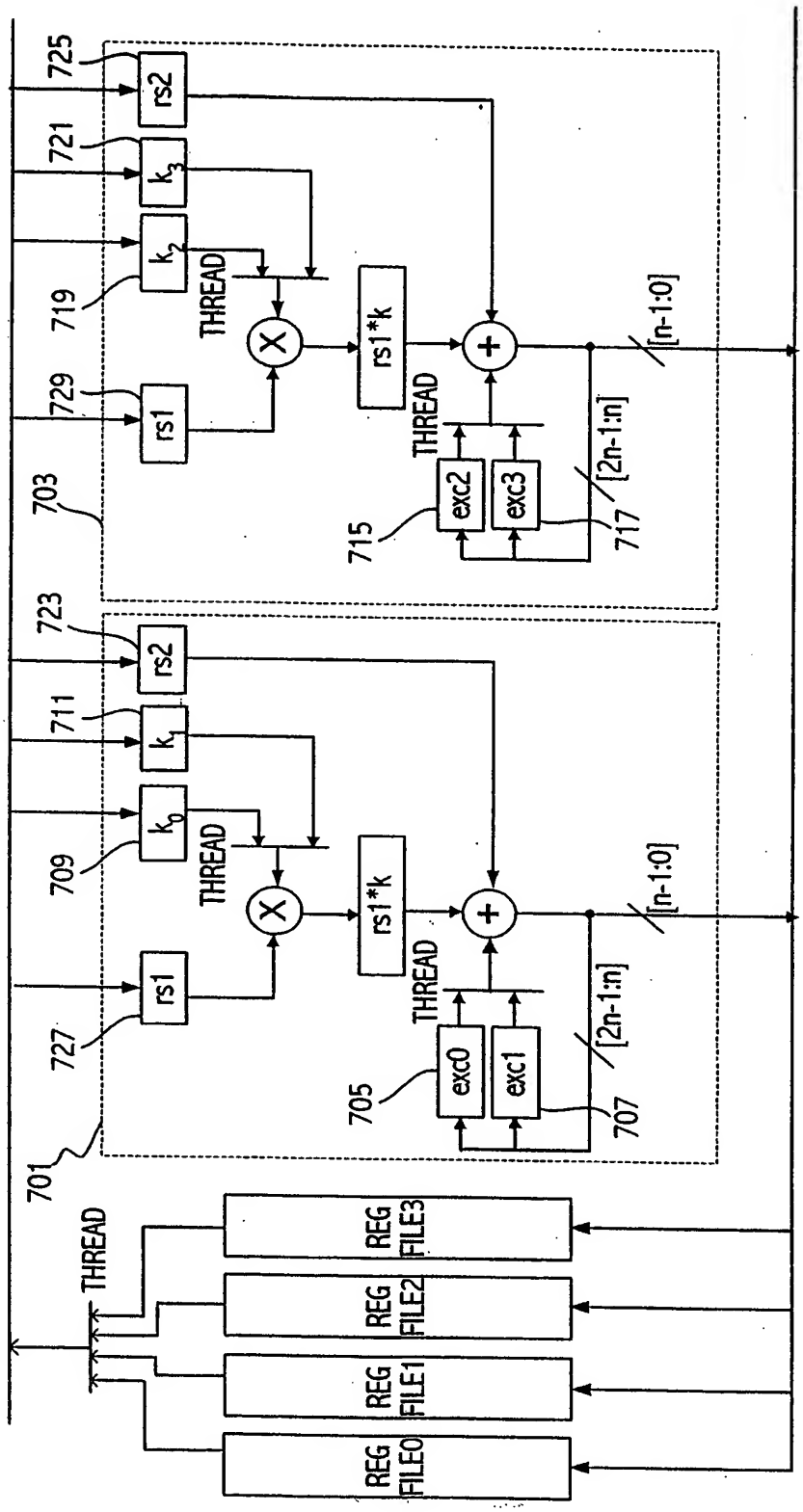

701
703
705
707
709
711
715
717
719
721
723
725
727
729
THREAD
rs1
rs2
k0
k1
k2
k3
rs1*k
exc0
exc1
exc2
exc3
[n-1:0]
[2n-1:n]
REG FILE0
REG FILE1
REG FILE2
REG FILE3

FIG 7

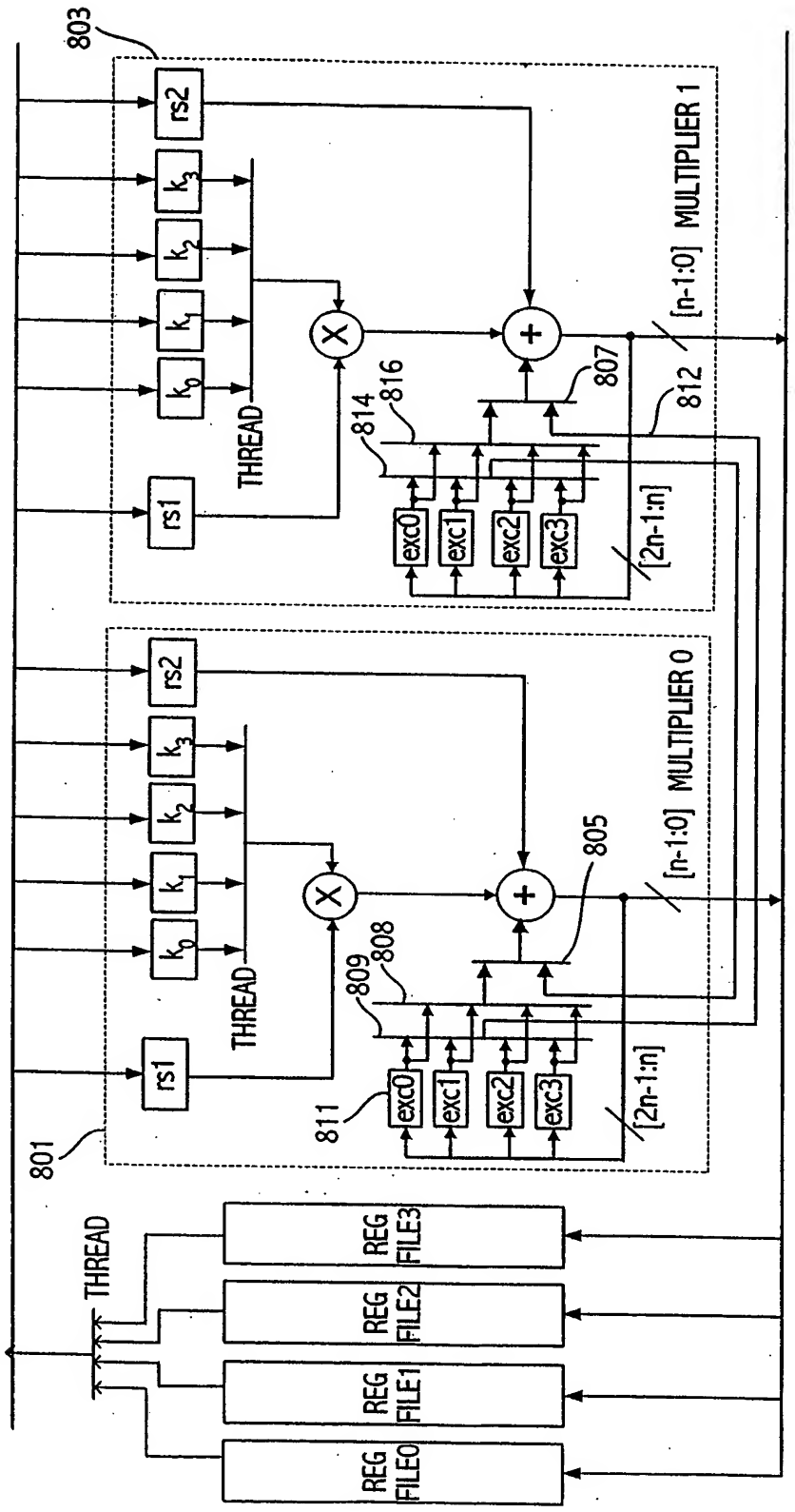

803
rs2
k3
k2
k1
k0
rs1
THREAD
MULTIPLIER 1
[n-1:0]
807
812
814
816
exc0
exc1
exc2
exc3
[2n-1:n]
MULTIPLIER 0
805
808
809
811
801
REG FILE3
REG FILE2
REG FILE1
REG FILE0

FIG. 8

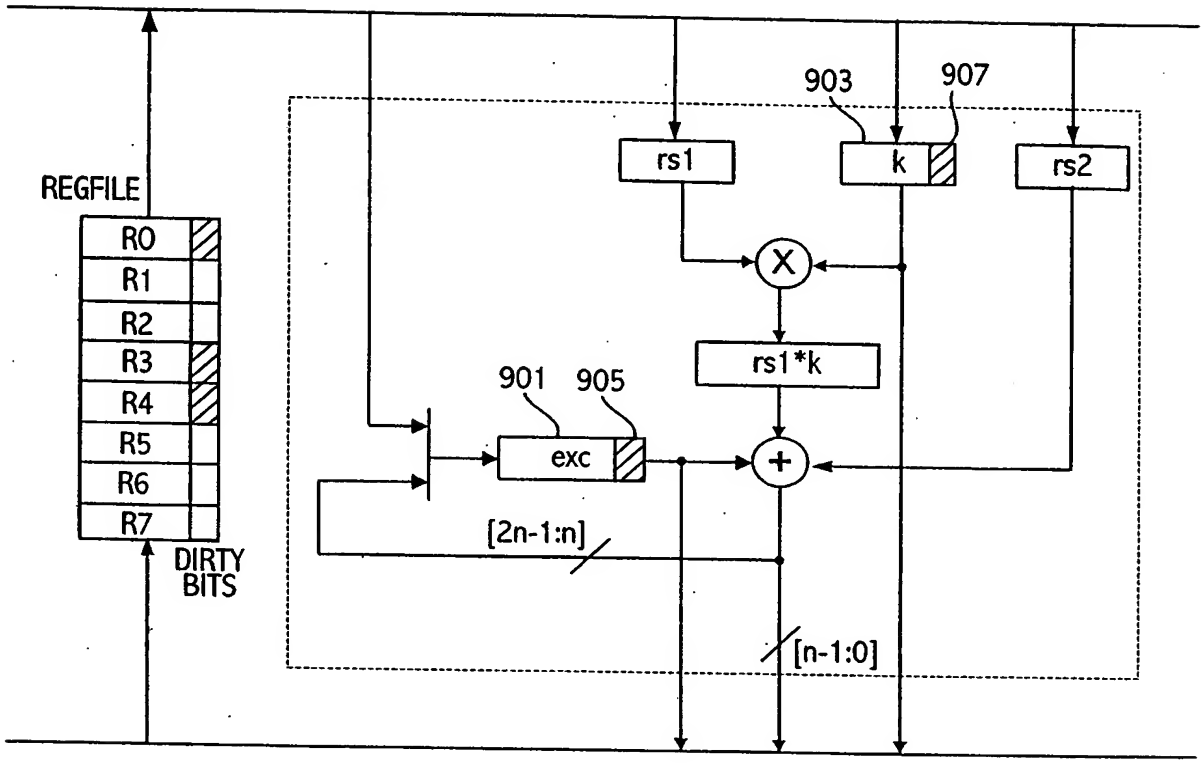

REGFILE
R0
R1
R2
R3
R4
R5
R6
R7
DIRTY BITS
903
907
rs1
k
rs2
X
rs1*k
901
905
exc
+
[2n-1:n]
[n-1:0]

FIG. 9

```
X3  X2  X1  X0  *  Y3  Y2  Y1  Y0  + EX3 EX2 EX1 EX0
                   P03 P02 P01 P00        P_IJ
+              P13 P12 P11 P10
+          P23 P22 P21 P20            = X_I * Y_J
+      P33 P32 P31 P30
+                  EX3 EX2 EX1 EX0
-------------------------------------
=      S6  S5  S4  S3  S2  S1  S0
+ C7   C6  C5  C4  C3  C2  C1
-------------------------------------
  EX3 EX2 EX1 EX0 rd3 rd2 rd1 rd0
```

FIG. 10

```
X3  X2  X1  X0  *  Y3  Y2  Y1  Y0 + (0,S6,S5,S4)+ (C7,C6,C5,C4) +(0,0,0,CC4)
                   P03 P02 P01 P00 ┐
+              P13 P12 P11 P10     │ INPUTS AT START
+          P23 P22 P21 P20         │ OF WALLACE TREE      P_ij = X_i * Y_j
+      P33 P32 P31 P30             ┘
+                  S6  S5  S4  ← PREVIOUS SUM OUTPUT   ┐ INPUTS IN MIDDLE
+                  C7  C6  C5  C4 ← PREVIOUS CARRY OUTPUT┘ OF WALLACE TREE
=  ---------------------------------
     S6  S5  S4  S3  S2  S1  S0  } NEW WALLACE TREE SUM OUTPUT
+ C7  C6  C5  C4  C3  C2  C1     } NEW WALLACE TREE CARRY OUTPUT
```

FIG. 11

```
          S3  S2  S1  S0
      +   C3  C2  C1
      +               CC4   ← PREVIOUS CLA CARRY OUT
  ---------------------------
      CC4 RD3 RD2 RD1 RD0   ← NEW CLA OUTPUT
```

FIG. 12

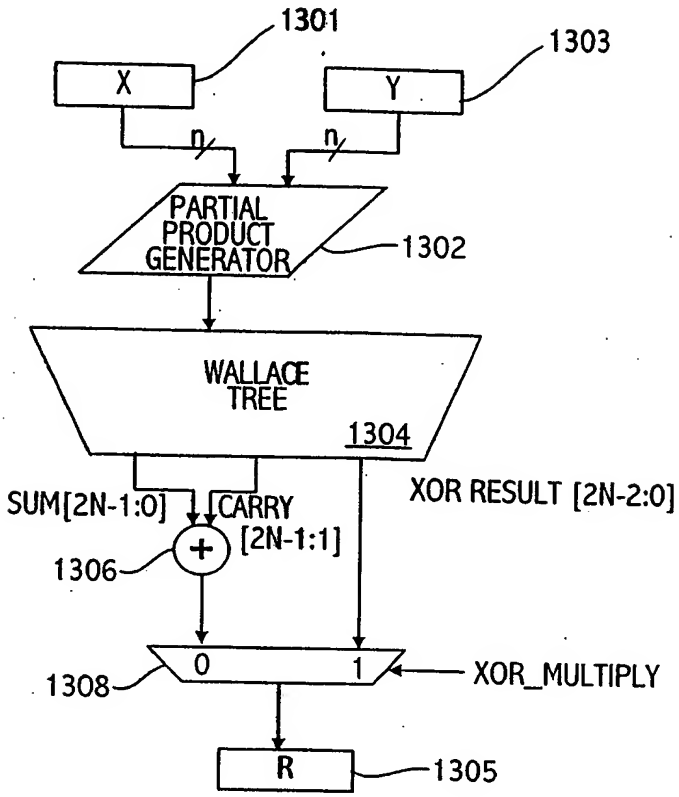


FIG. 13

```
                                ~SD   SD   SD   D8   D7   D6   D5   D4   D3   D2   D1   D0
                              1 ~SE   E8   E7   E6   E5   E4   E3   E2   E1   E0        SD
                         1 ~SF  F8   F7   F6   F5   F4   F3   F2   F1   F0        SE
                    1 ~SG  G8   G7   G6   G5   G4   G3   G2   G1   G0        SF
                      H7   H6   H5   H4   H3   H2   H1   H0        SG
                                                        S15  S14  S13  S12  S11  S10  S9   S8
C16  C16  C16  C16  C16  C16  C16  C16  C16  C16  C15  C14  C13  C12  C11  C10  C9   C8
                                                        Z7   Z6   Z5   Z4   Z3   Z2   Z1   Z0
-----  ---------------------------------------------------------------------------------------
     S15  S14  S13  S12  S11  S10  S9   S8   S7   S6   S5   S4   S3   S2   S1   S0
C16  C15  C14  C13  C12  C11  C10  C9   C8   C7   C6   C5   C4   C3   C2   C1   C0

with [S7:S0] and [C7:C0] going to the carry look-ahead adder.
```

Fig 12B

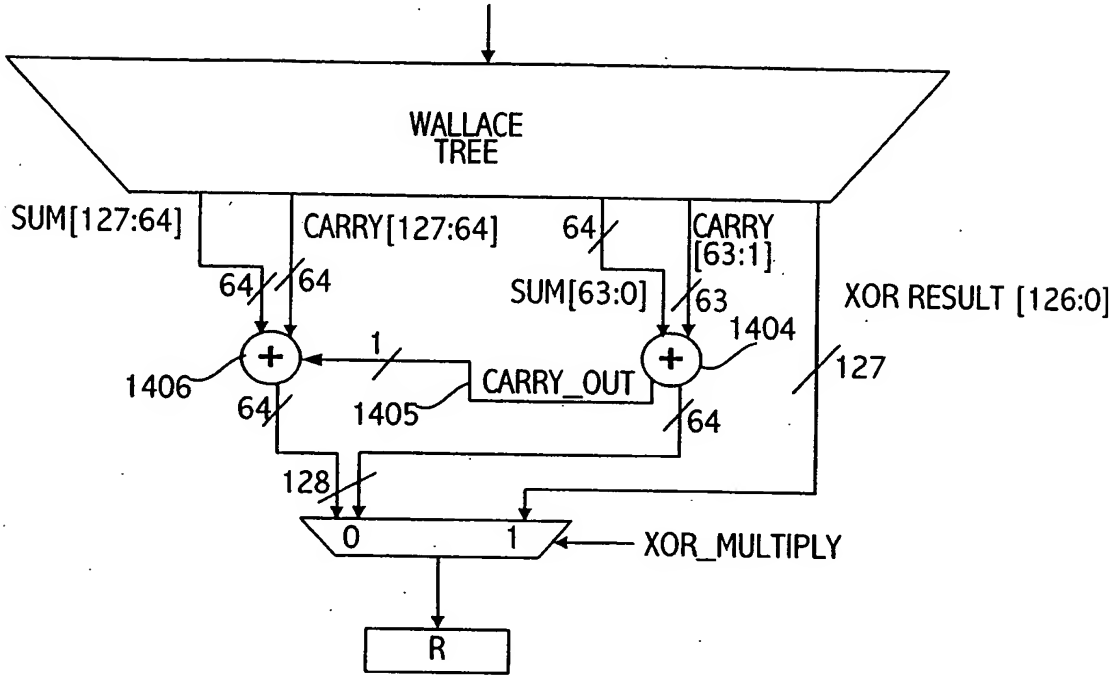

WALLACE
TREE
SUM[127:64]
CARRY[127:64]
64
CARRY
[63:1]
64
64
SUM[63:0]
63
XOR RESULT [126:0]
1404
1
1406
CARRY_OUT
127
64
1405
64
128
0
1
XOR_MULTIPLY
R

FIG. 14

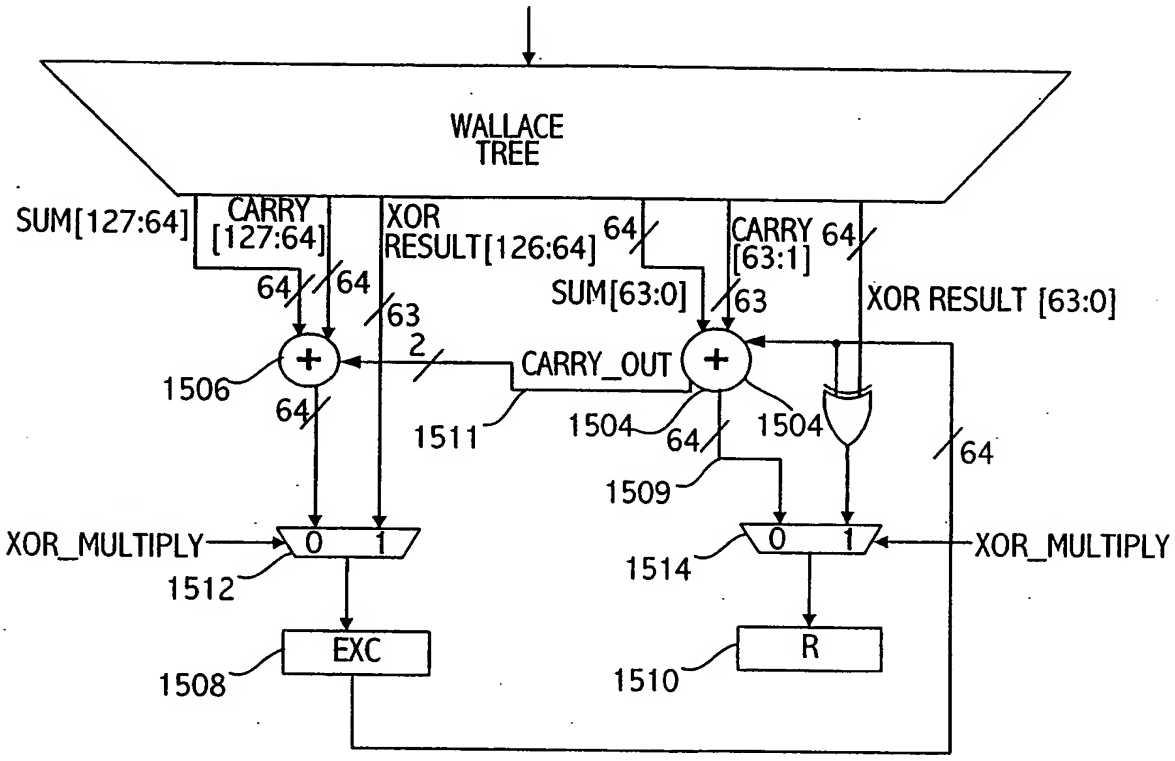

WALLACE TREE
SUM[127:64]
CARRY [127:64]
XOR RESULT[126:64]
64
CARRY [63:1]
64
SUM[63:0]
63
XOR RESULT [63:0]
64
64
63
2
CARRY_OUT
1506
64
1511
1504
64
1504
64
1509
XOR_MULTIPLY
0
1
1512
1514
0
1
XOR_MULTIPLY
EXC
1508
R
1510

FIG. 15

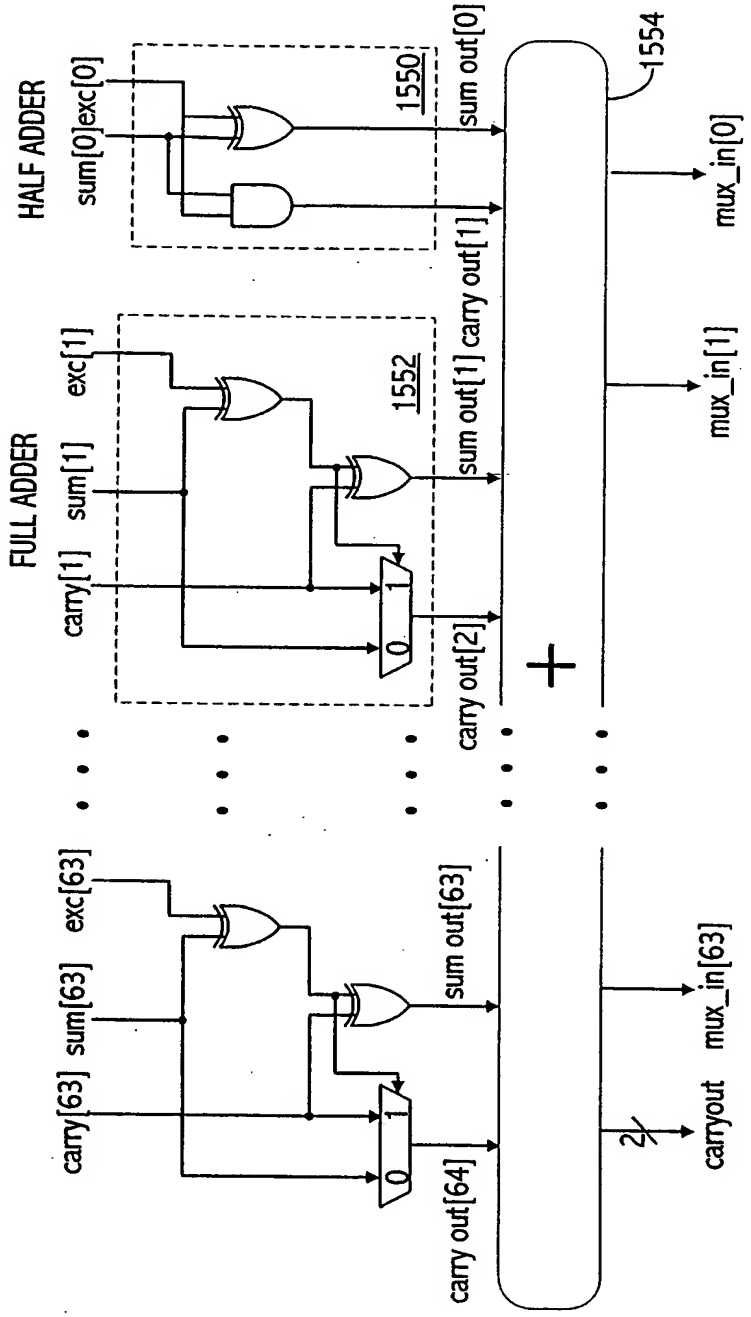

HALF ADDER
sum[0]exc[0]
1550
sum out[0]
1554
mux_in[0]
FULL ADDER
exc[1]
sum[1]
carry[1]
1552
sum out[1] carry out[1]
mux_in[1]
carry out[2]
exc[63]
sum[63]
carry[63]
sum out[63]
mux_in[63]
carryout
carry out[64]
2

FIG. 15a

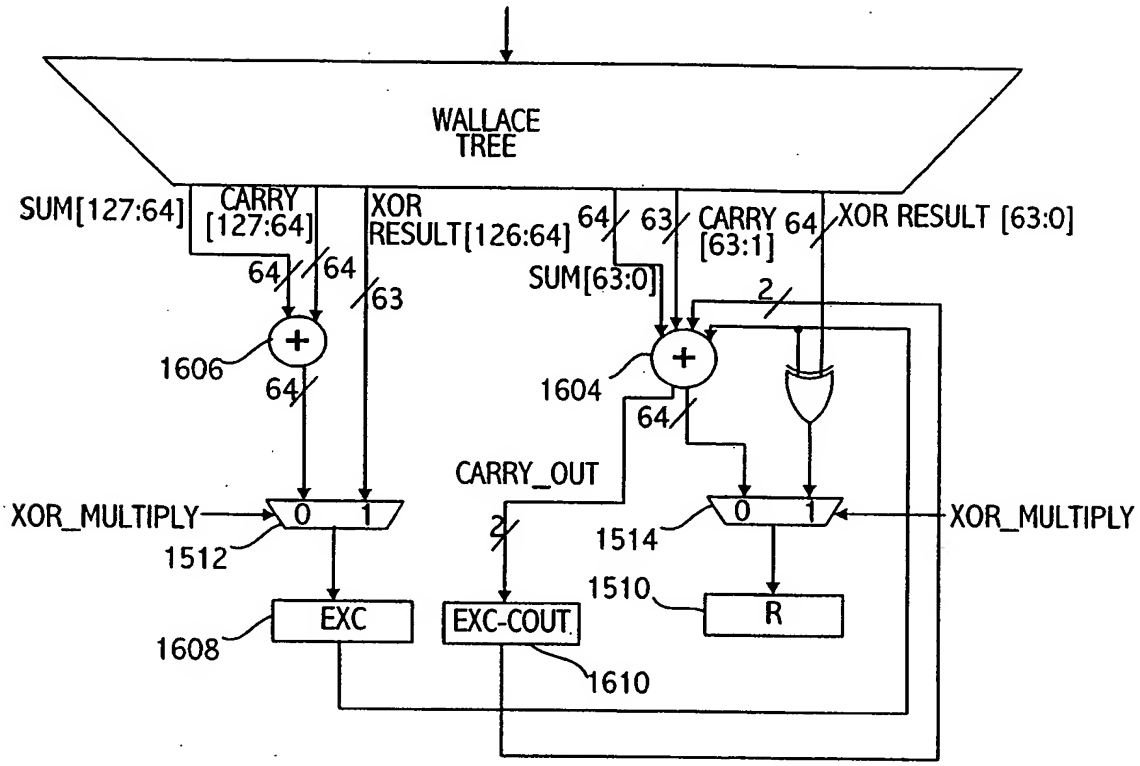

WALLACE TREE
SUM[127:64]
CARRY [127:64]
XOR RESULT[126:64]
64
63
CARRY [63:1]
64
XOR RESULT [63:0]
64
64
63
SUM[63:0]
2
1606
1604
64
64
CARRY_OUT
XOR_MULTIPLY
0
1
2
1514
0
1
XOR_MULTIPLY
1512
1510
EXC
EXC-COUT
R
1608
1610

FIG. 16

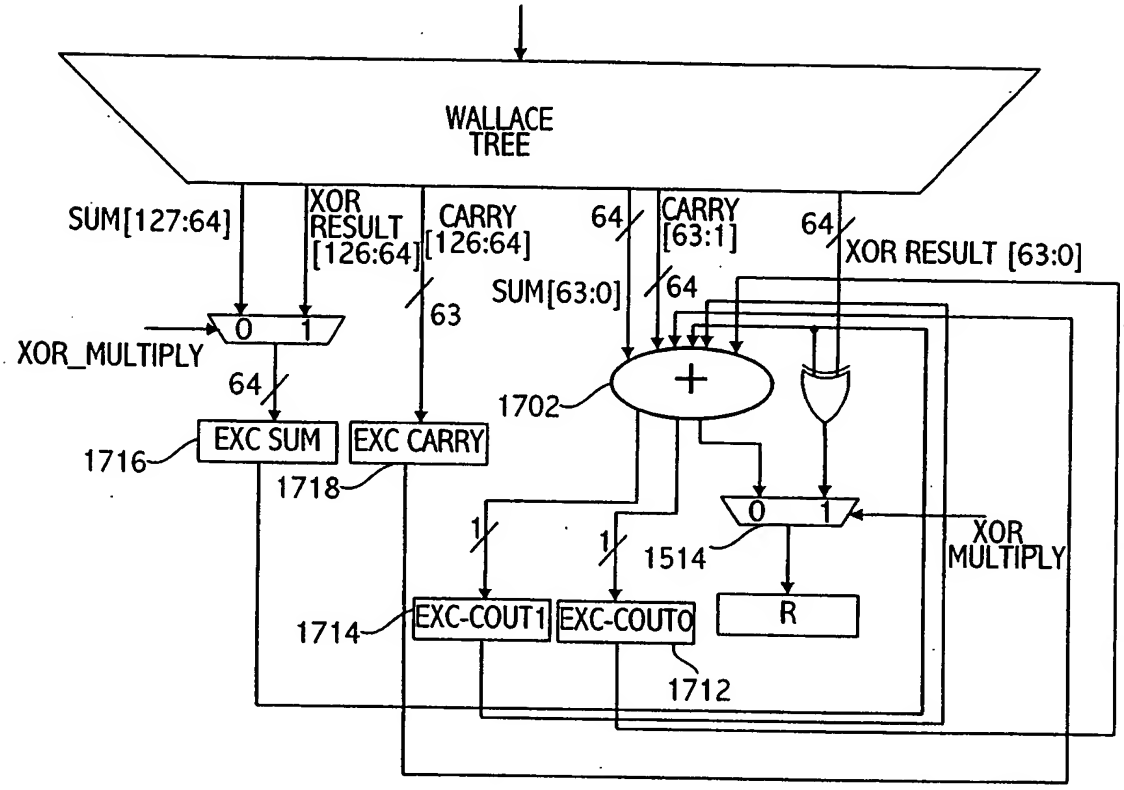

WALLACE TREE
SUM[127:64]
XOR RESULT [126:64]
CARRY [126:64]
64
CARRY [63:1]
64
XOR RESULT [63:0]
SUM[63:0]
64
63
XOR_MULTIPLY
0
1
64
1702
+
EXC SUM
EXC CARRY
1716
1718
0
1
XOR MULTIPLY
1
1
1514
EXC-COUT1
EXC-COUT0
R
1714
1712

FIG. 17

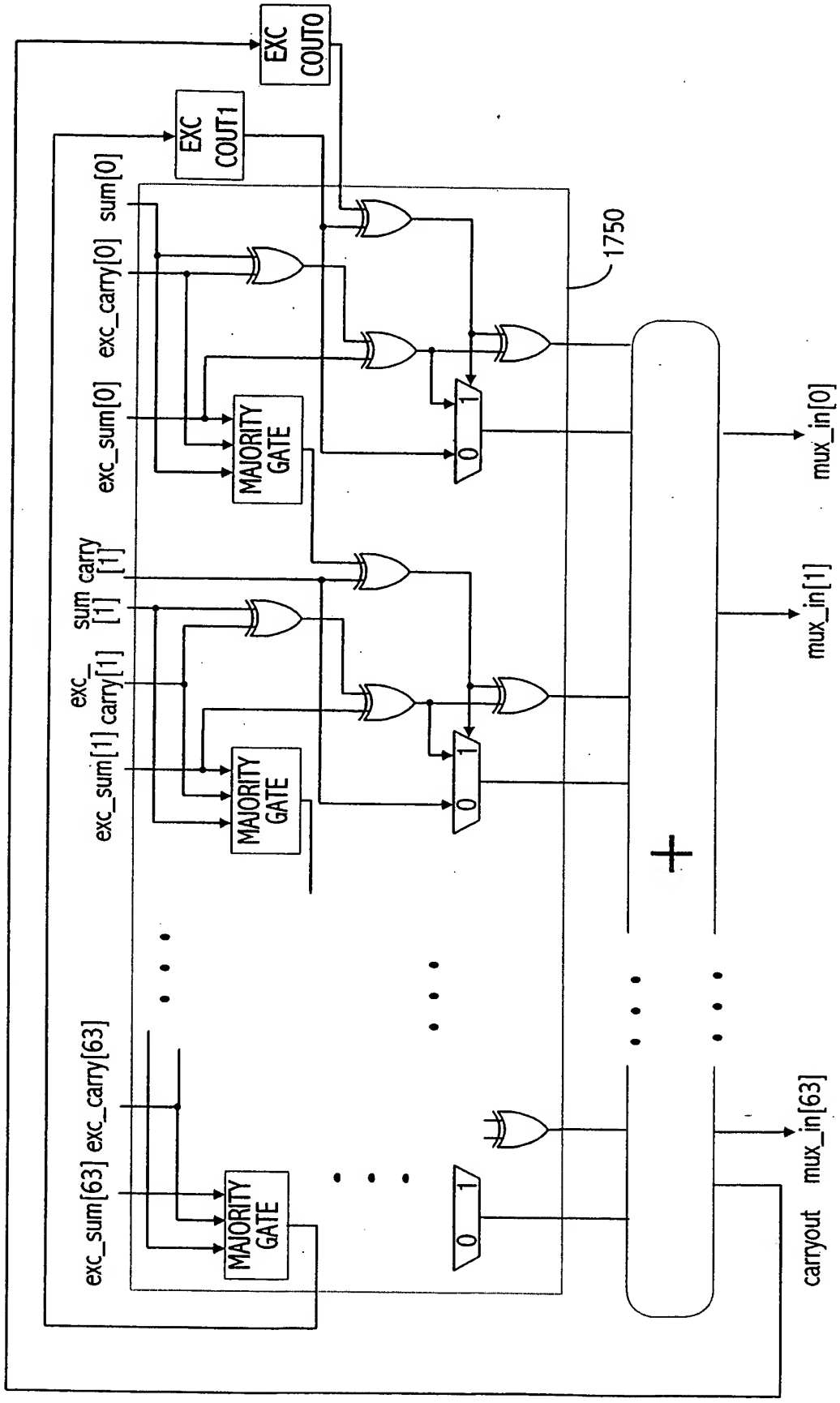

EXC COUT0
EXC COUT1
1750
sum[0]
exc_carry[0]
exc_sum[0]
MAJORITY GATE
sum carry [1]
exc_ carry[1]
exc_sum[1]
MAJORITY GATE
exc_sum[63]
exc_carry[63]
MAJORITY GATE
0 1
mux_in[0]
mux_in[1]
mux_in[63]
carryout
+

Fig. 17A

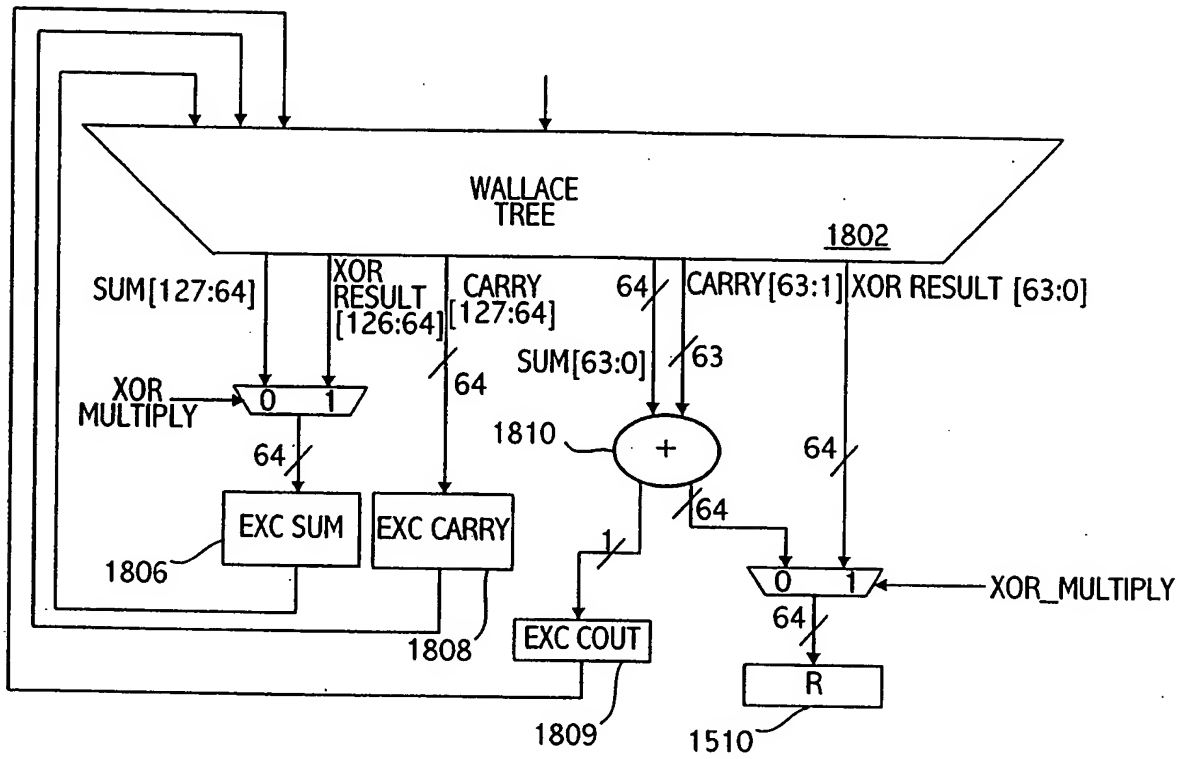

WALLACE TREE
1802
SUM[127:64]
XOR RESULT [126:64]
CARRY [127:64]
64
CARRY[63:1]
XOR RESULT [63:0]
SUM[63:0]
63
XOR MULTIPLY
0
1
64
1810
+
64
64
EXC SUM
EXC CARRY
1806
1
64
0
1
XOR_MULTIPLY
1808
EXC COUT
64
R
1809
1510

FIG. 18

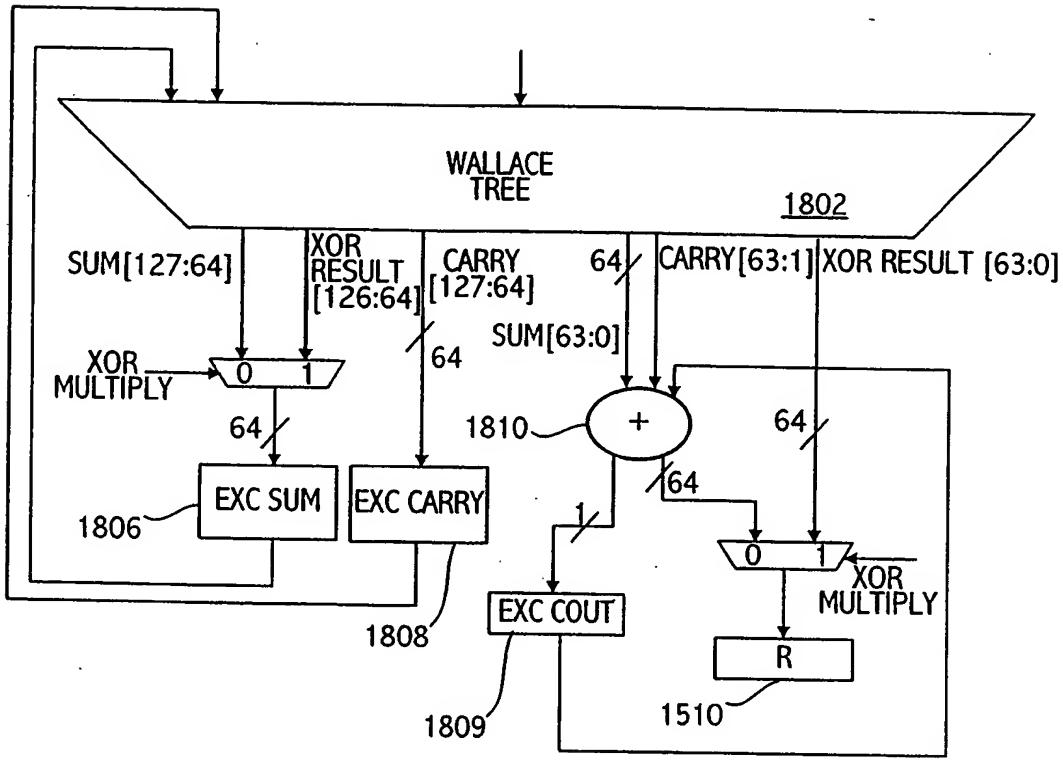

WALLACE TREE
1802
SUM[127:64]
XOR RESULT [126:64]
CARRY [127:64]
64
CARRY[63:1]
XOR RESULT [63:0]
SUM[63:0]
XOR MULTIPLY
0
1
64
64
1810
+
64
EXC SUM
EXC CARRY
1806
64
1
0
1
XOR MULTIPLY
EXC COUT
1808
R
1809
1510

FIG. 19

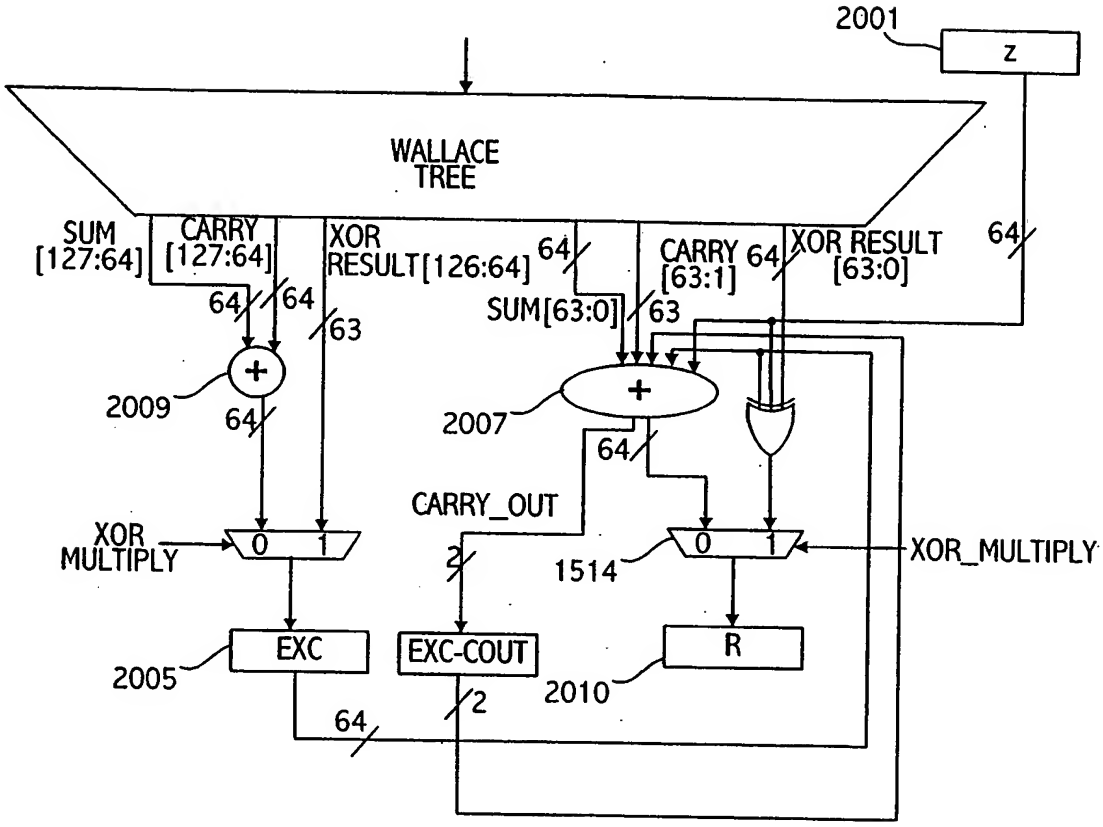

2001
Z
WALLACE TREE
SUM [127:64]
CARRY [127:64]
XOR RESULT[126:64]
64
CARRY [63:1]
64
XOR RESULT [63:0]
64
64
64
63
SUM[63:0]
63
2009
2007
64
64
CARRY_OUT
XOR MULTIPLY
0
1
2
1514
0
1
XOR_MULTIPLY
2005
EXC
EXC-COUT
R
2010
64
2

FIG. 20

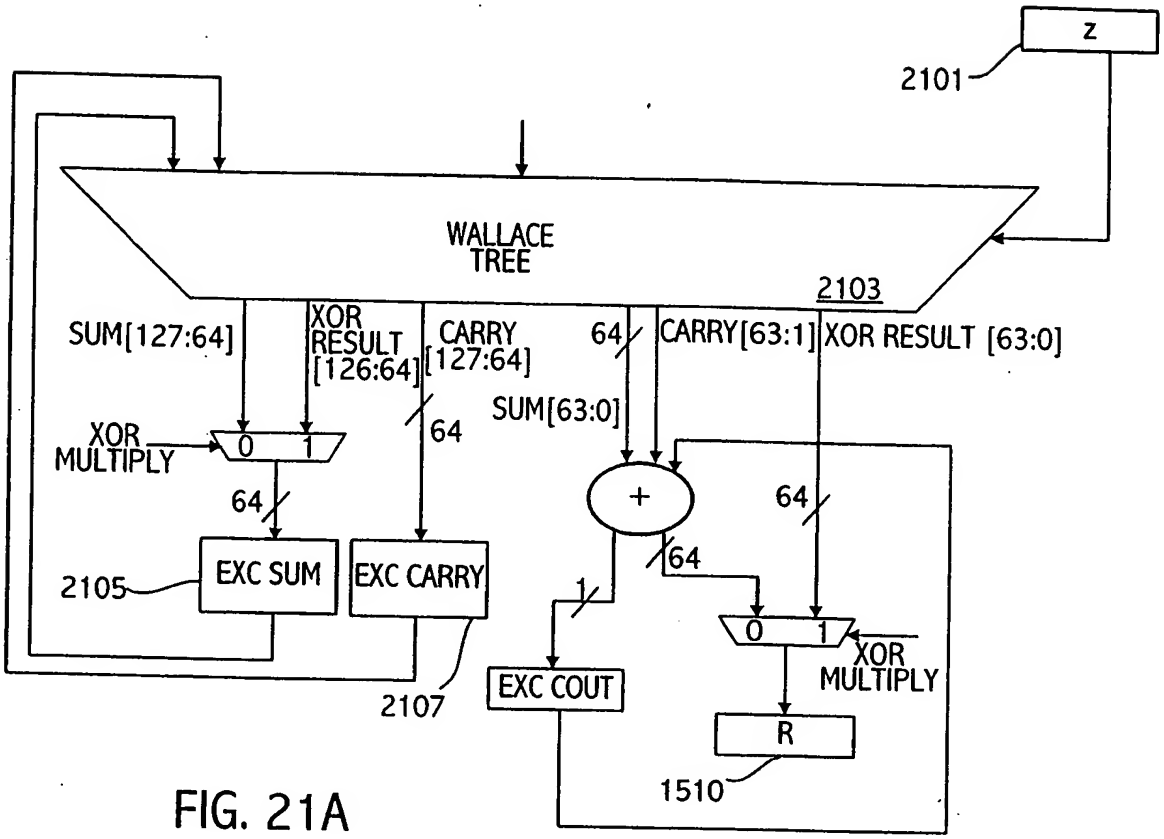


FIG. 21A

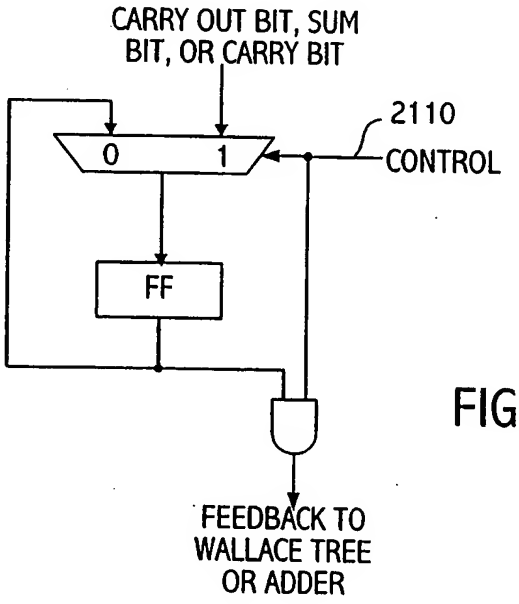


FIG. 21B

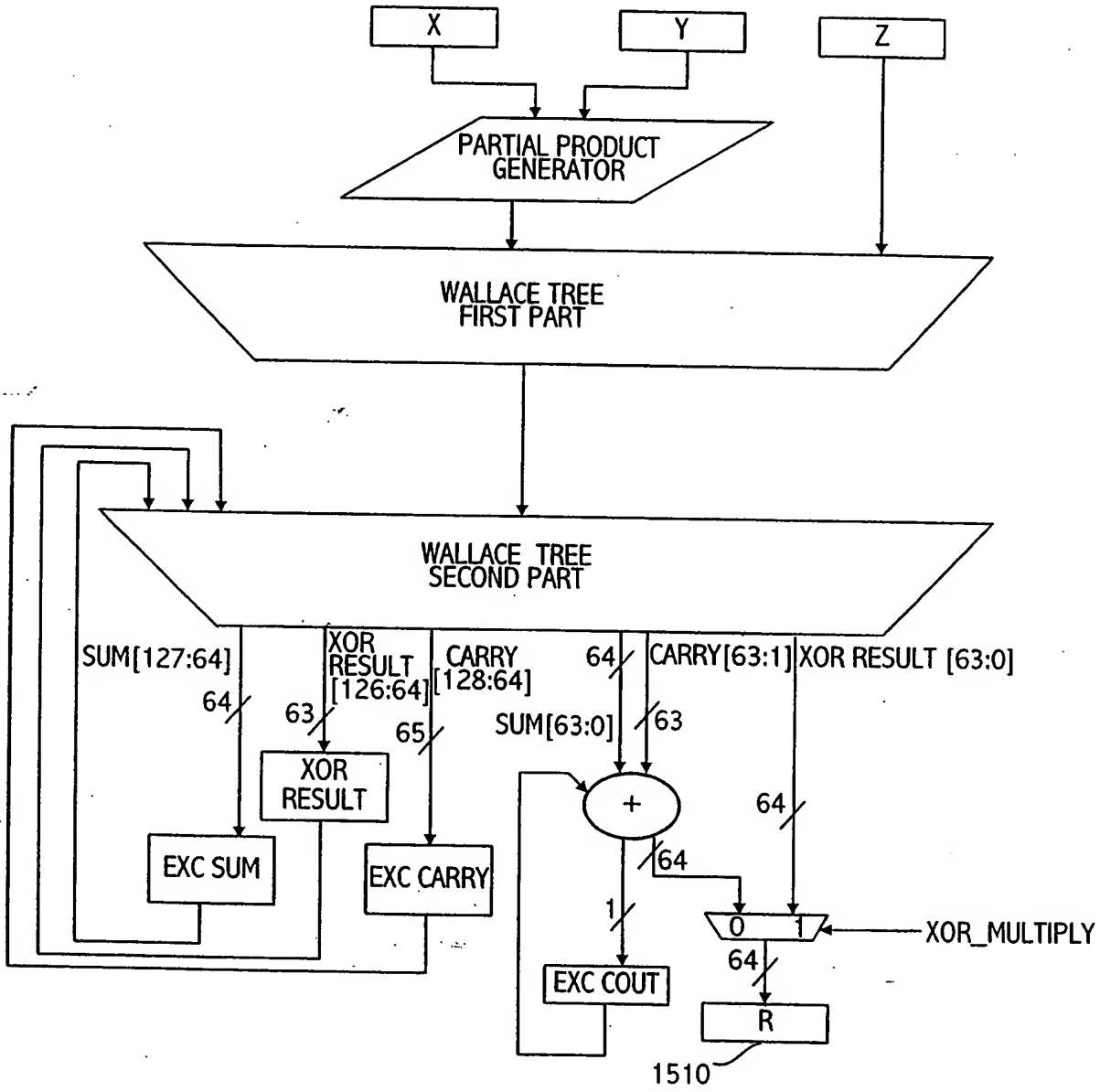
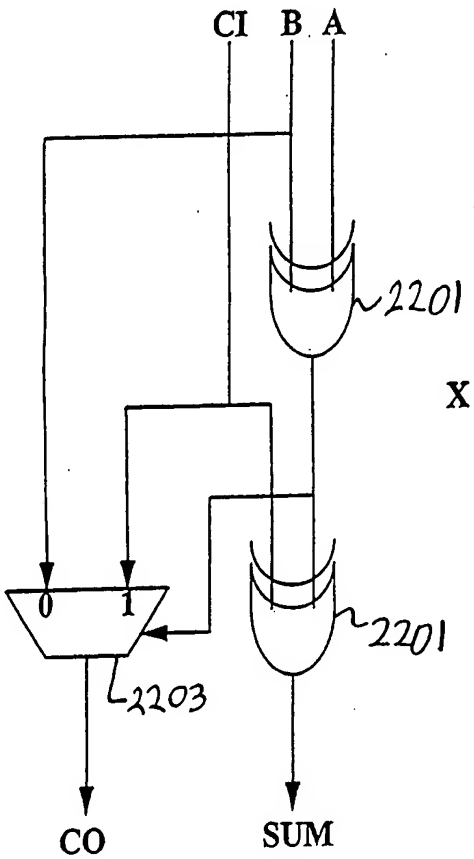

X
Y
Z
PARTIAL PRODUCT GENERATOR
WALLACE TREE FIRST PART
WALLACE TREE SECOND PART
SUM[127:64]
64
XOR RESULT [126:64]
63
CARRY [128:64]
65
64
CARRY[63:1]
XOR RESULT [63:0]
SUM[63:0]
63
XOR RESULT
EXC SUM
EXC CARRY
+
64
64
1
0
1
XOR_MULTIPLY
64
EXC COUT
R
1510

FIG. 21C

FULL ADDER

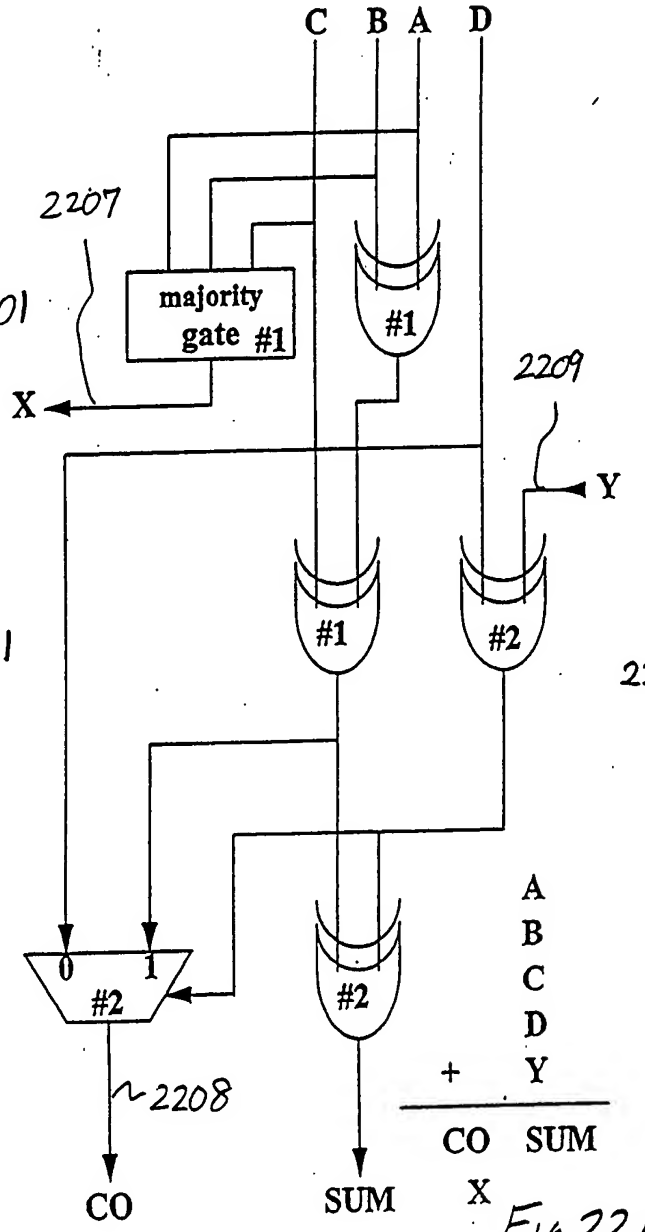


efficiency = 18.4%

```
    A
    B
+  CI
-------
CO  SUM
```

Fig. 22A

4 TO 2 COMPRESSOR

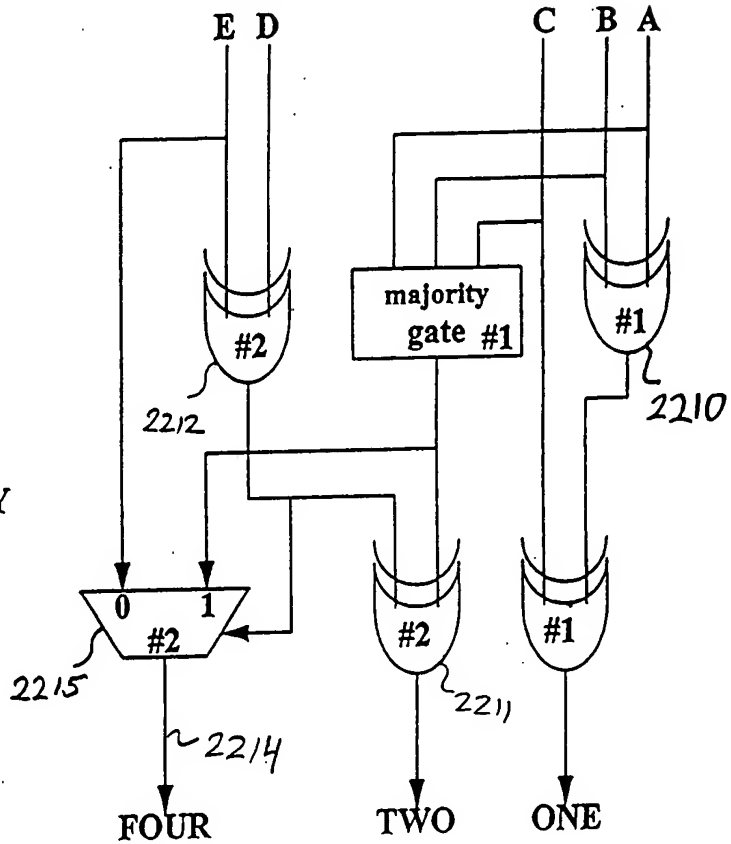


efficiency = 20.6%

```
    A
    B
    C
    D
+   Y
-------
CO  SUM
X
```

Fig 22B

5 TO 3 COMPRESSOR



efficiency = 22.5%

```
    A
  D B
+ E C
-------
4 2 1
```

Fig. 22C

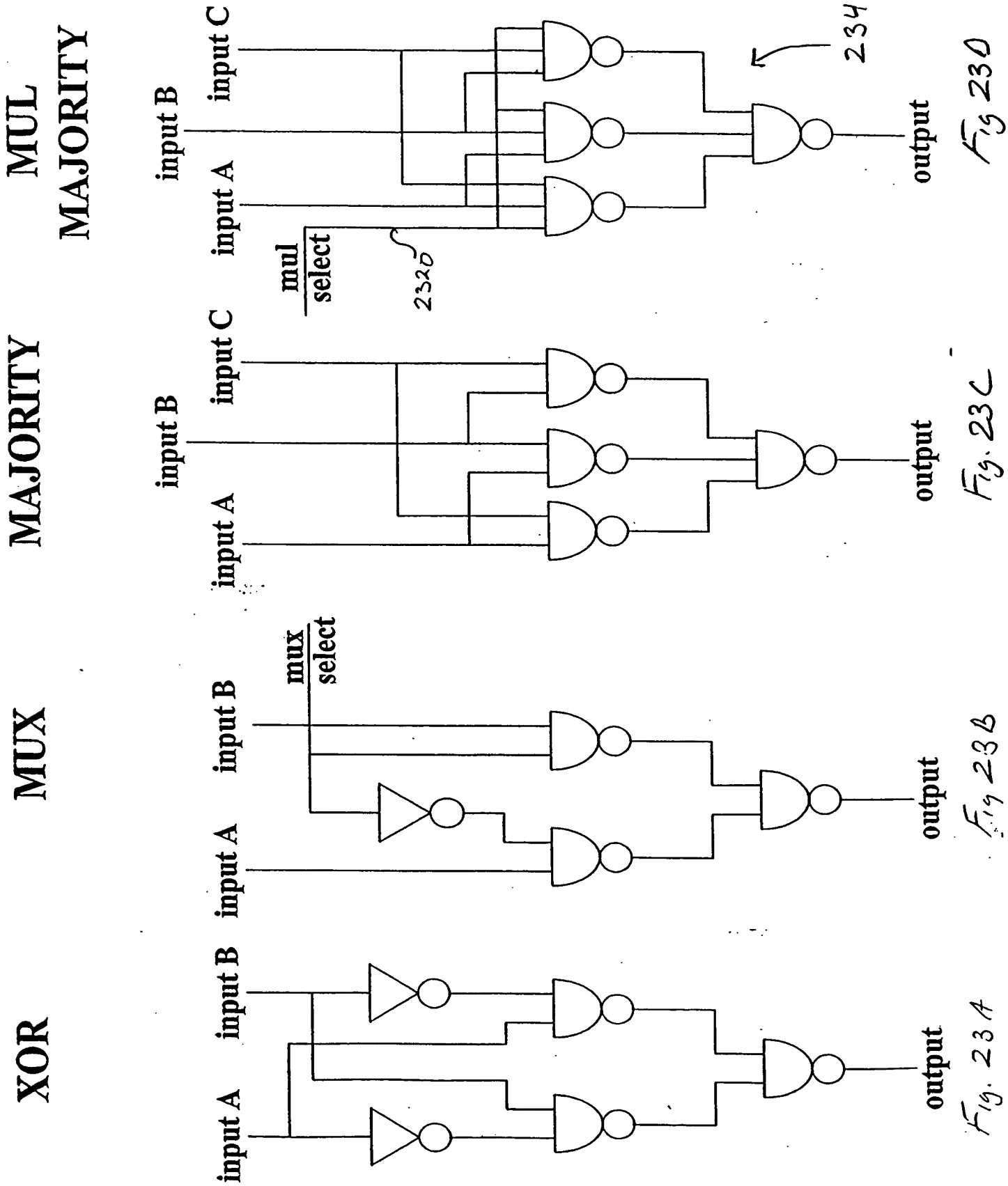


Fig. 234 Fig. 232a Fig. 232c Fig 230

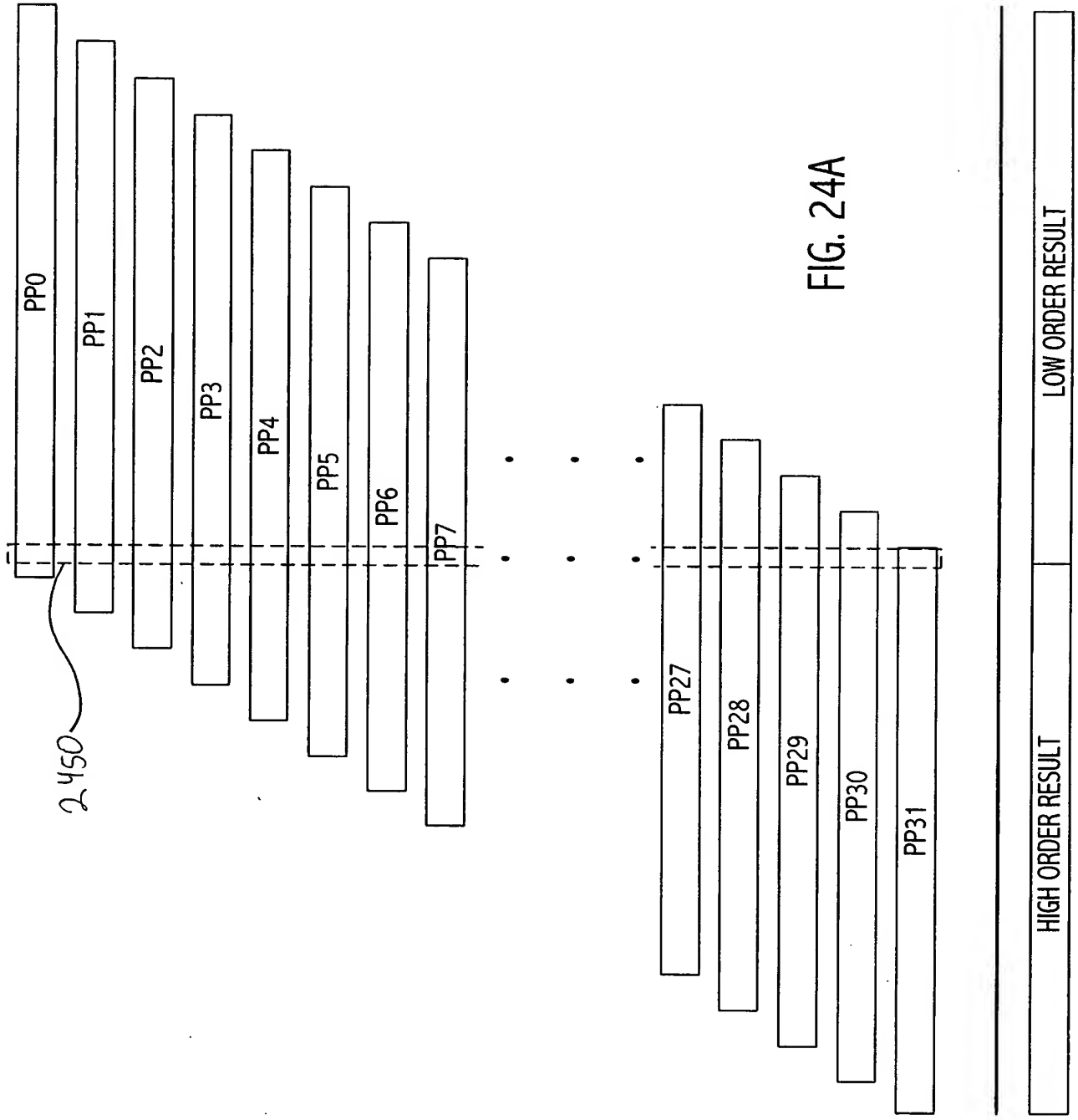

PP0
PP1
PP2
PP3
PP4
PP5
PP6
PP7
PP27
PP28
PP29
PP30
PP31
2450
HIGH ORDER RESULT
LOW ORDER RESULT

FIG. 24A

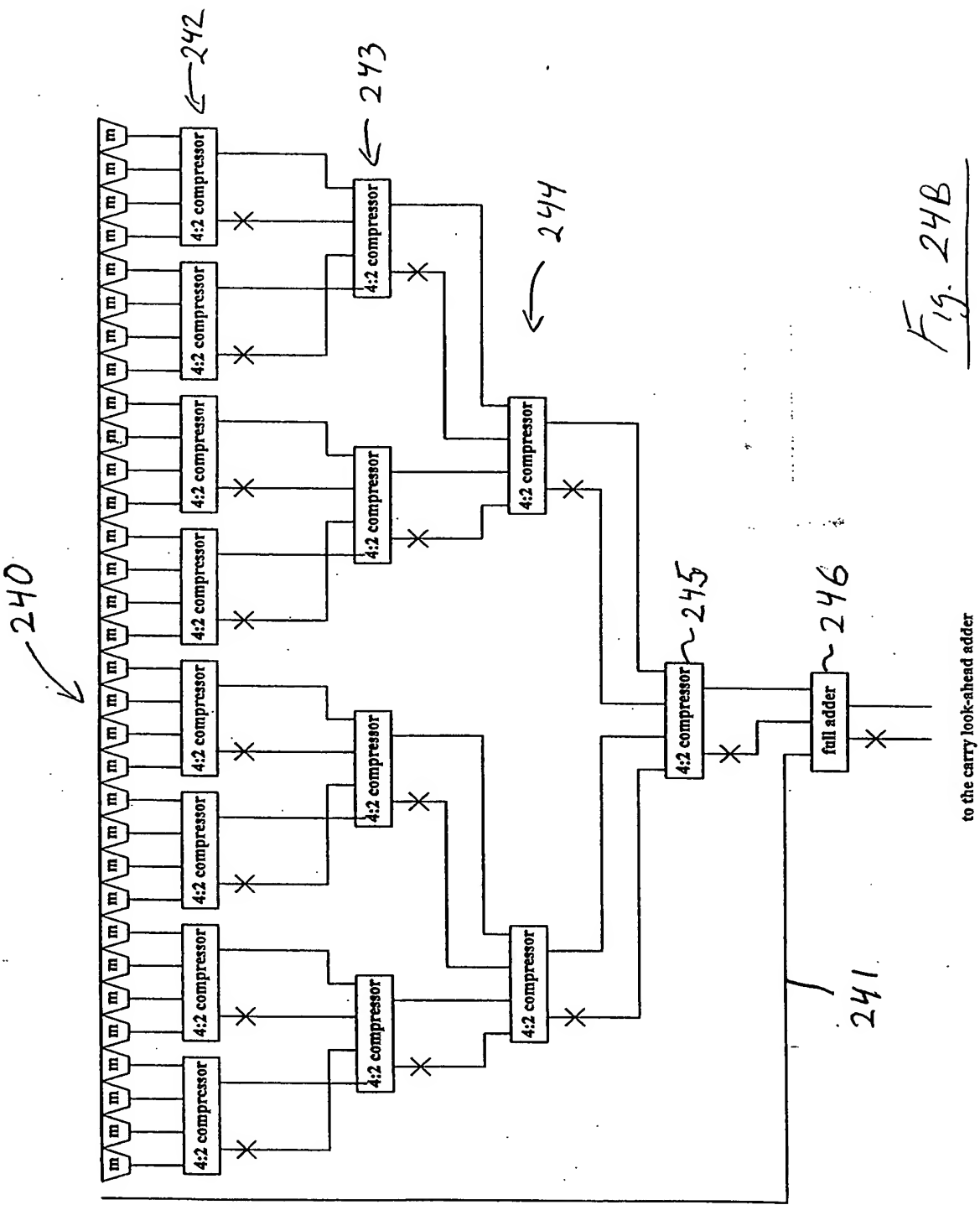

240
242
243
244
245
246
241
m
4:2 compressor
full adder
to the carry look-ahead adder

Fig. 24B

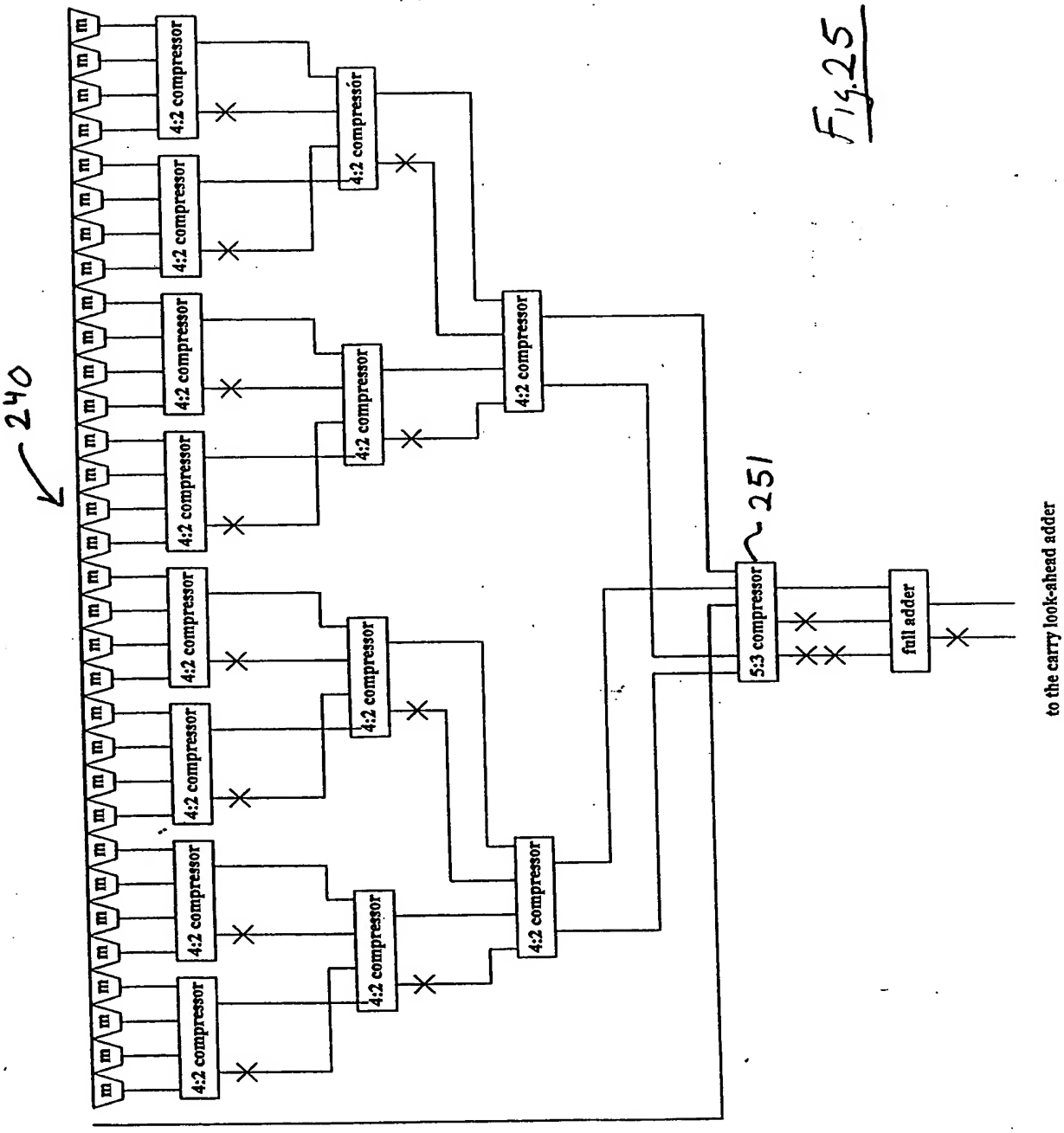

240
m
4:2 compressor
5:3 compressor
251
full adder
to the carry look-ahead adder

Fig. 25

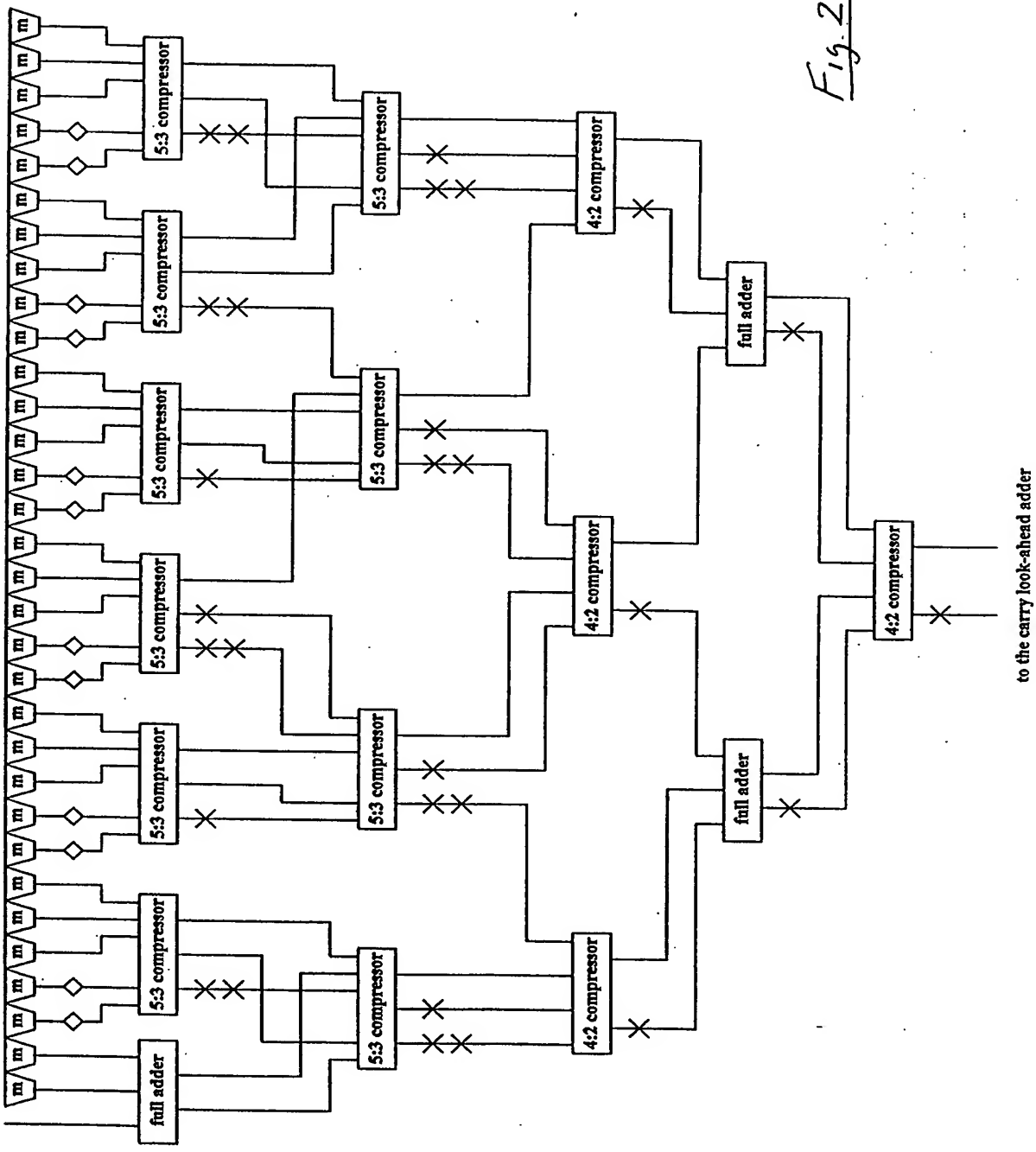

m
5:3 compressor
4:2 compressor
full adder
to the carry look-ahead adder

Fig. 26

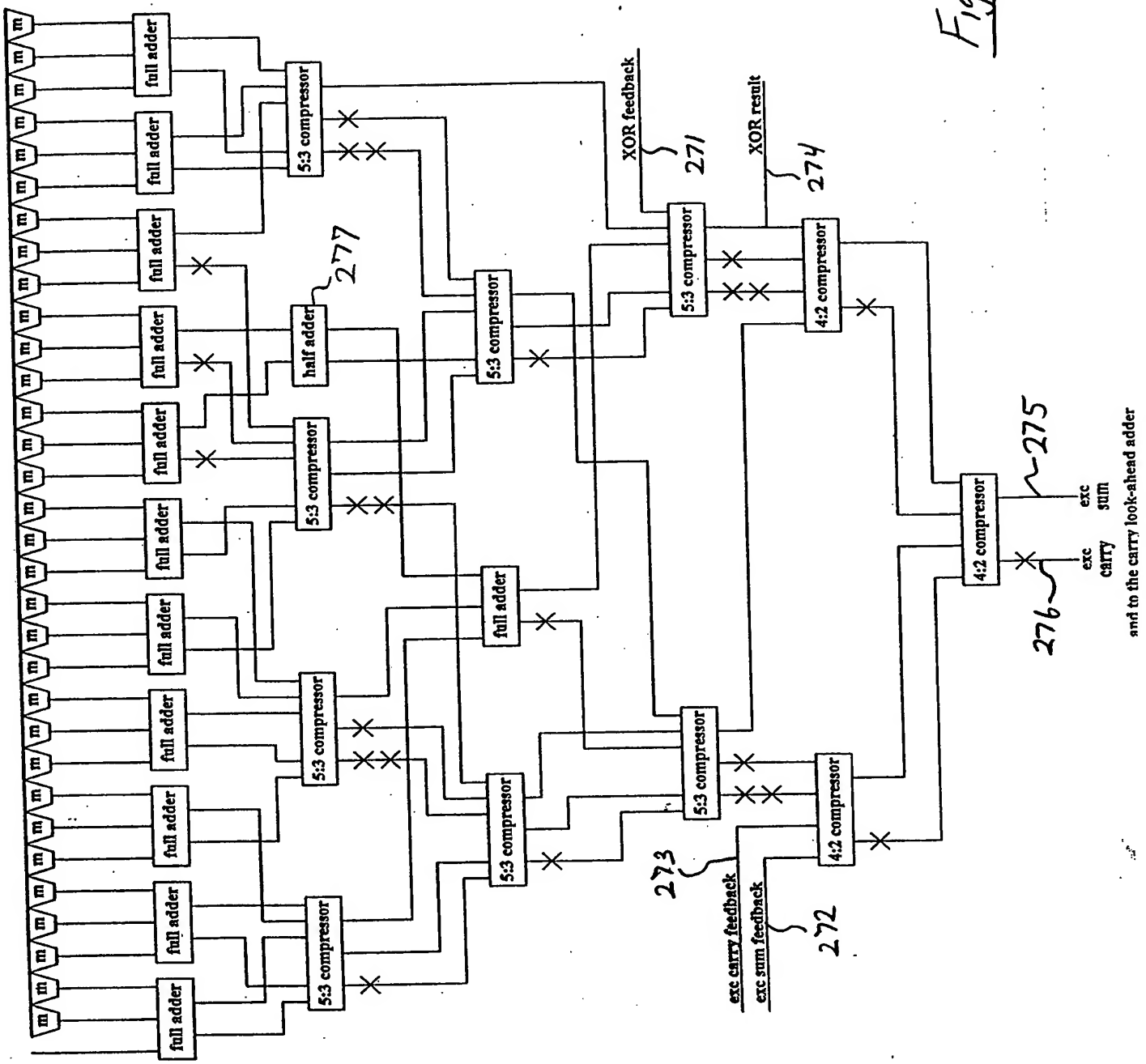

m
full adder
5:3 compressor
half adder
4:2 compressor
XOR feedback
XOR result
exc carry feedback
exc sum feedback
exc sum
exc carry
and to the carry look-ahead adder
271
272
273
274
275
276
277

Fig. 27

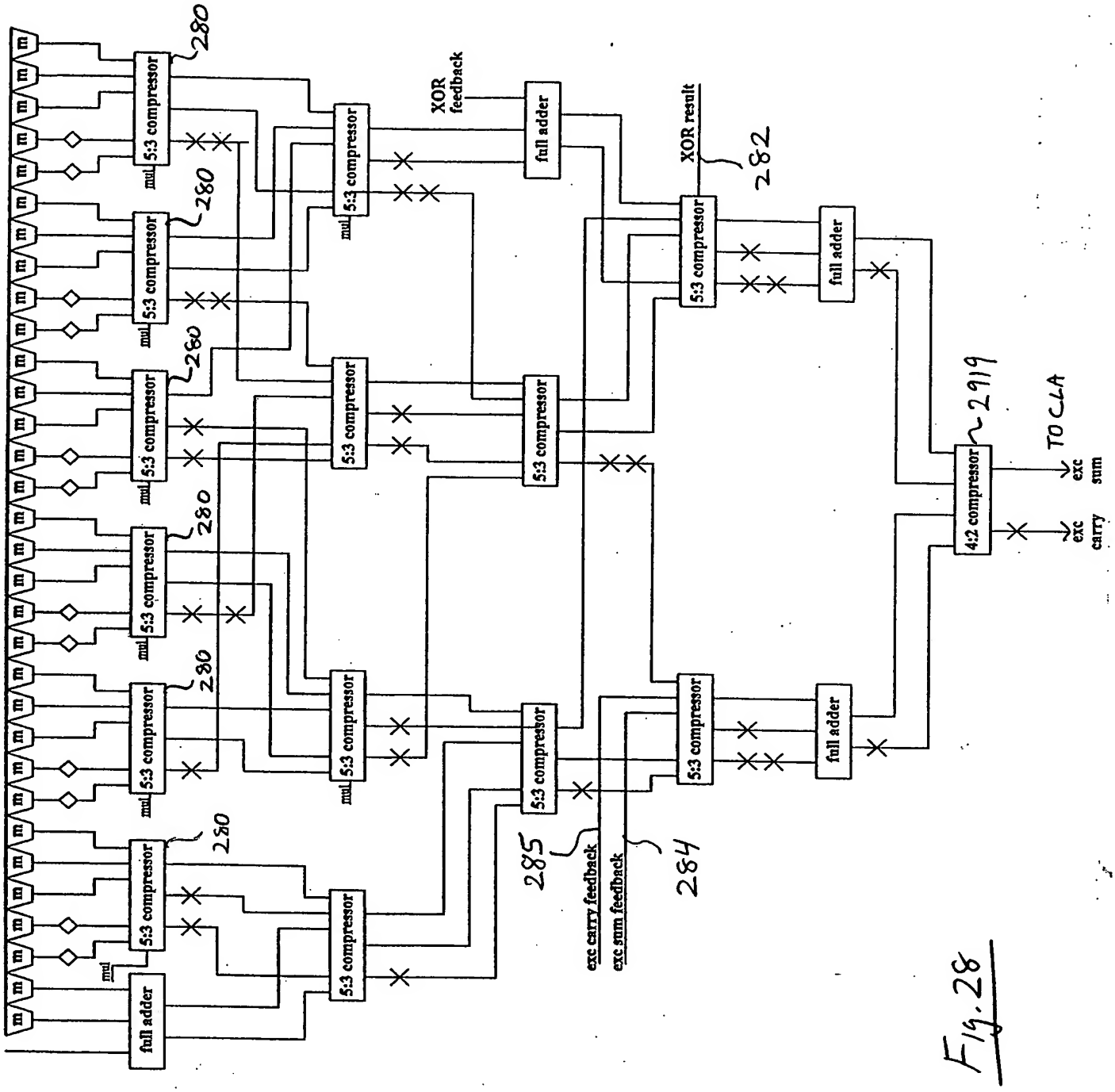

m
mul
5:3 compressor
full adder
4:2 compressor
280
282
284
285
291
XOR feedback
XOR result
exc carry feedback
exc sum feedback
TO CLA
exc sum
exc carry

Fig. 28

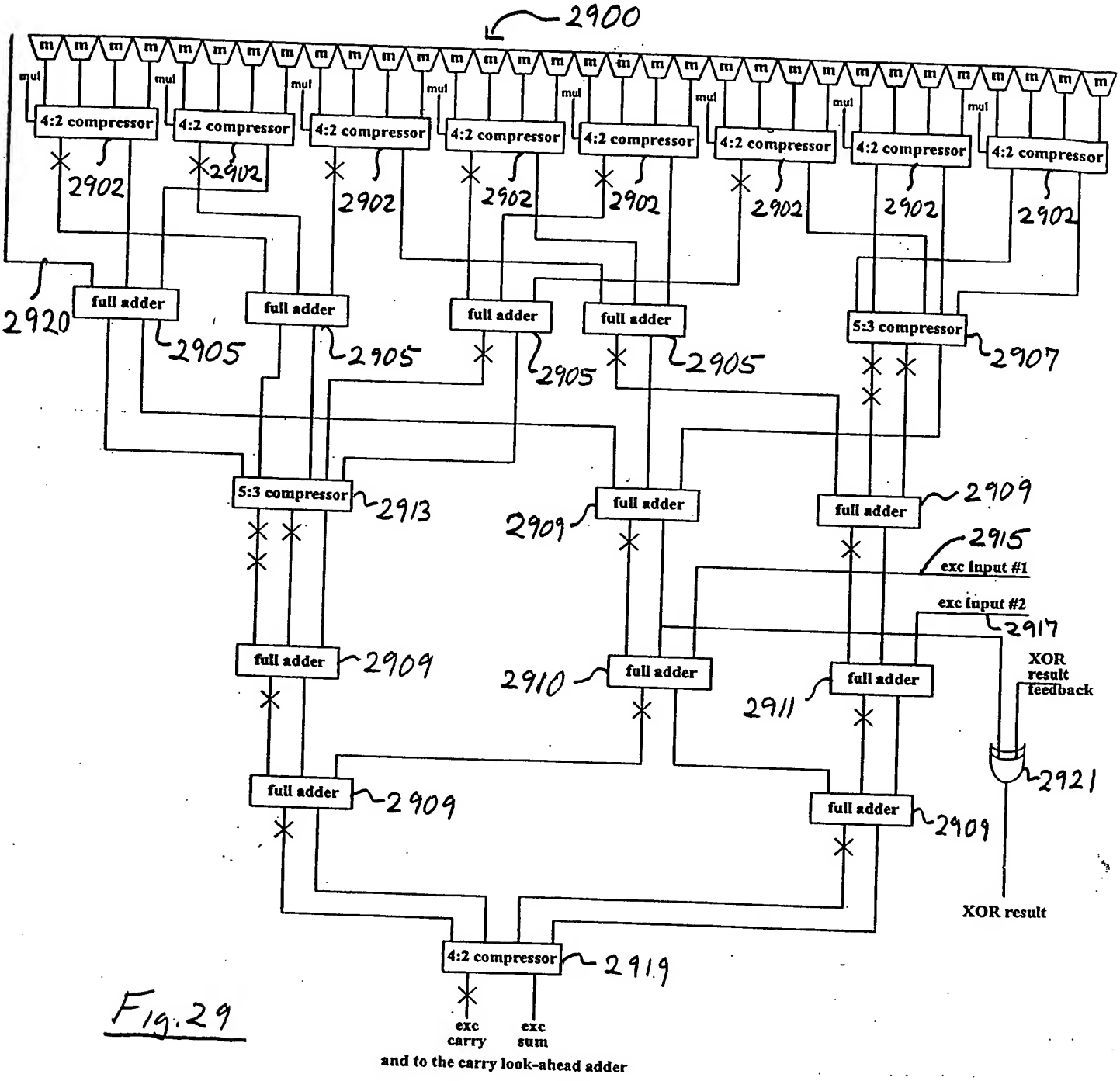

2900
m
mul
4:2 compressor
2902
full adder
2905
2920
5:3 compressor
2907
2913
2909
2915
exc input #1
exc input #2
2917
XOR result feedback
2910
2911
2921
XOR result
2919
exc carry
exc sum
and to the carry look-ahead adder

Fig. 29

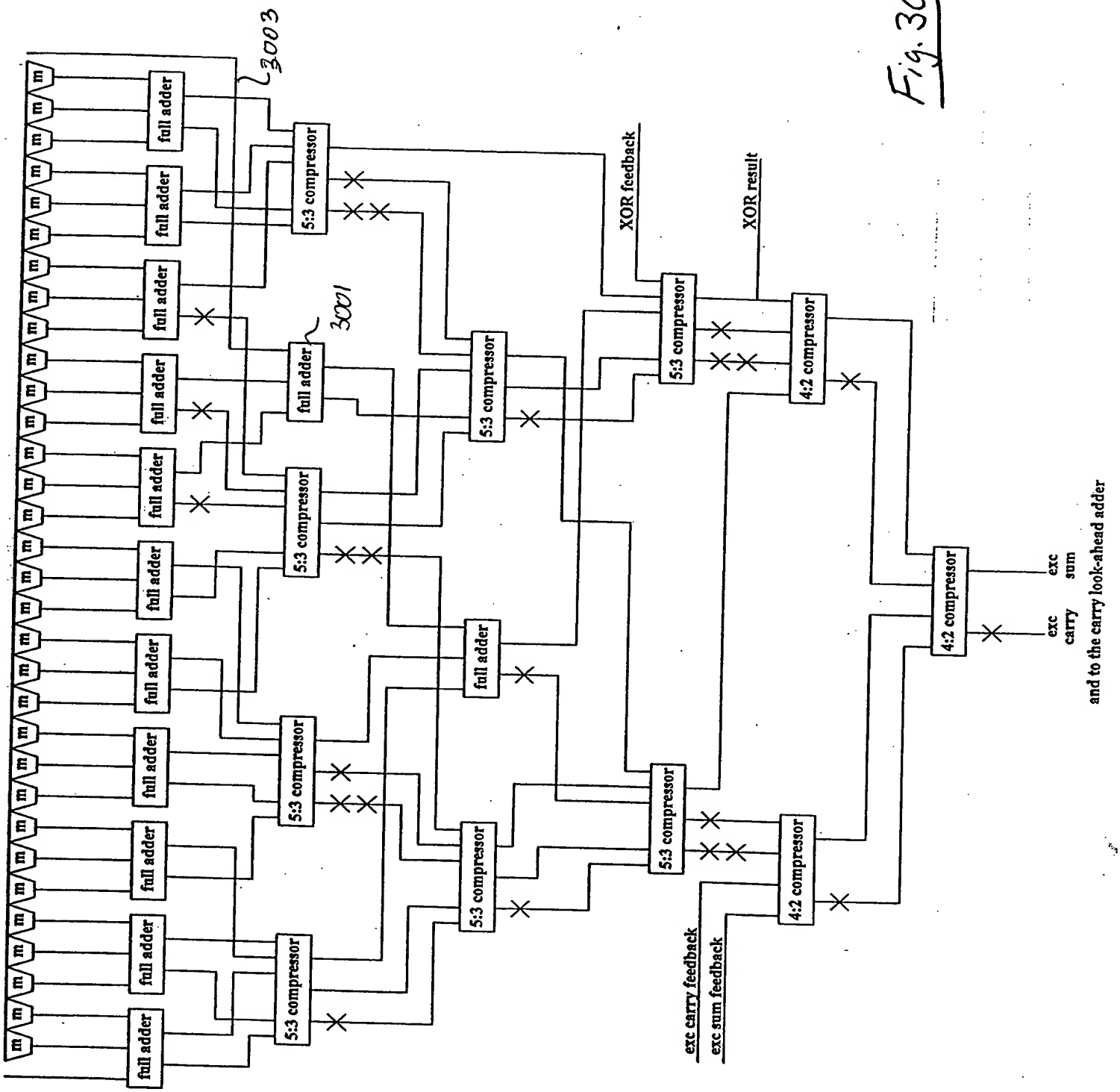

3003
3001
full adder
5:3 compressor
4:2 compressor
XOR feedback
XOR result
exc carry feedback
exc sum feedback
exc sum
exc carry
and to the carry look-ahead adder

Fig. 30

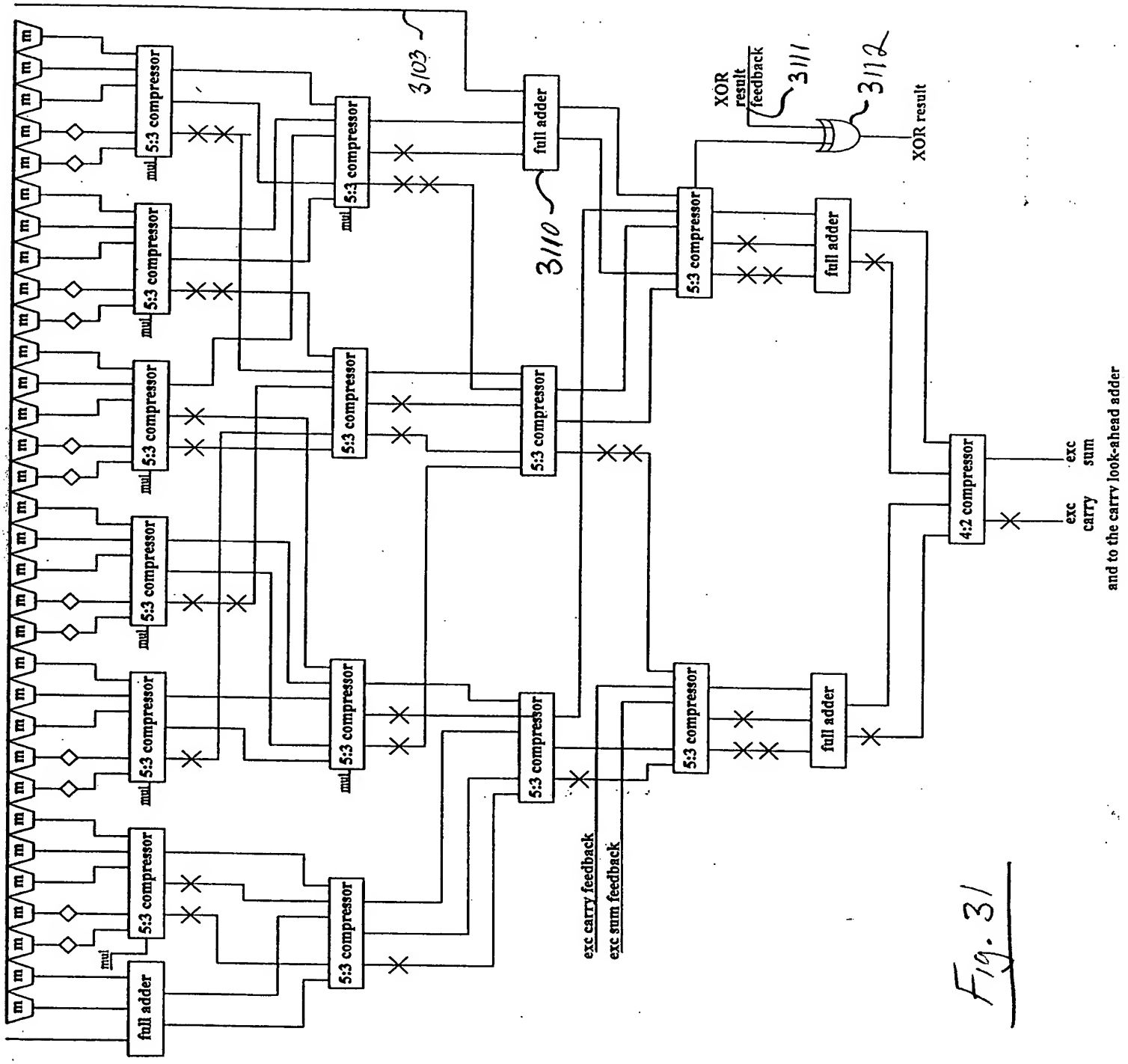

3103
3110
3111
3112
XOR result feedback
XOR result
5:3 compressor
full adder
4:2 compressor
mul
m
exc carry feedback
exc sum feedback
exc sum
exc carry
and to the carry look-ahead adder

Fig. 31

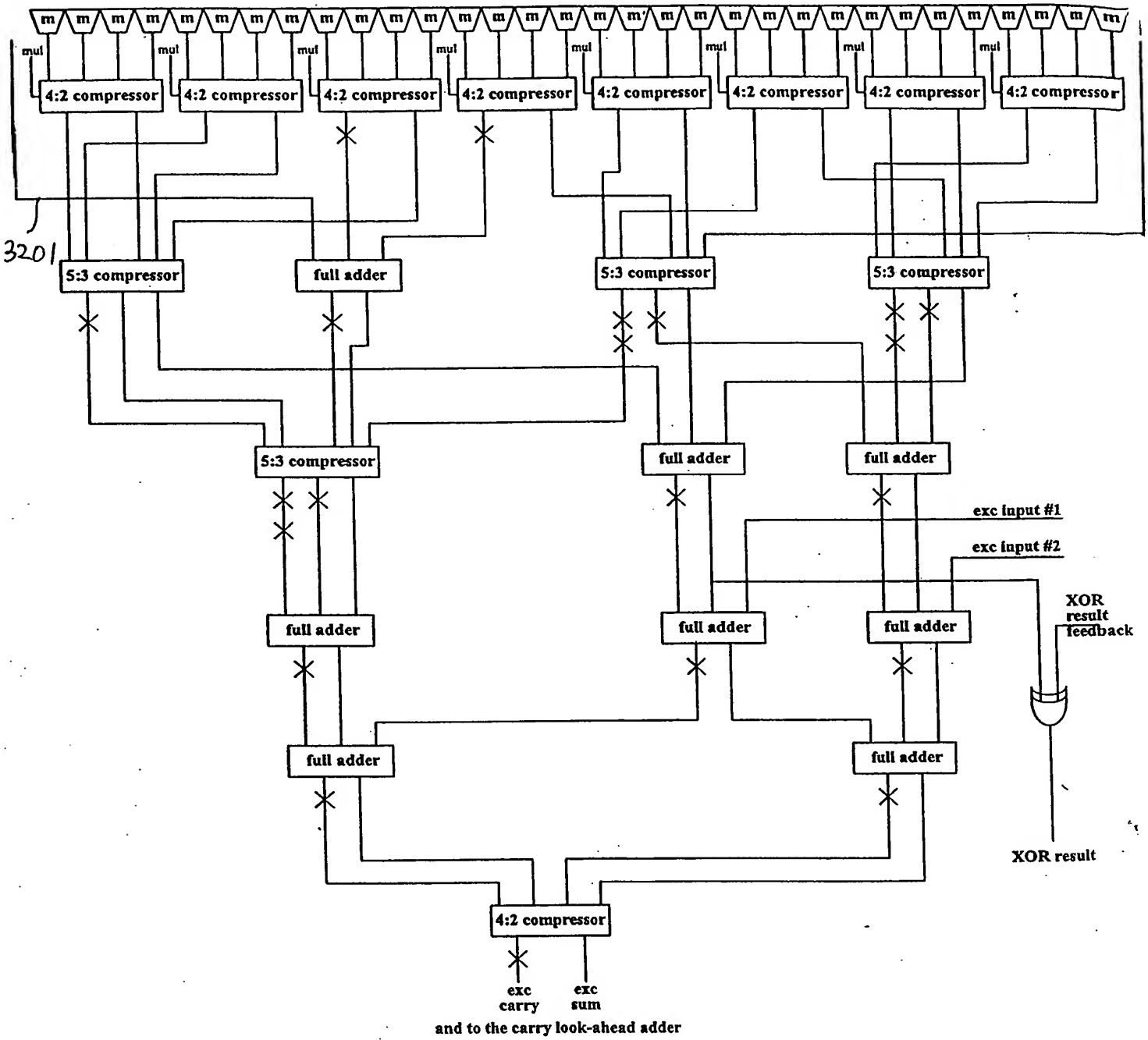

m
mul
4:2 compressor
3201
5:3 compressor
full adder
exc input #1
exc input #2
XOR result feedback
XOR result
exc carry
exc sum
and to the carry look-ahead adder

Fig. 32